

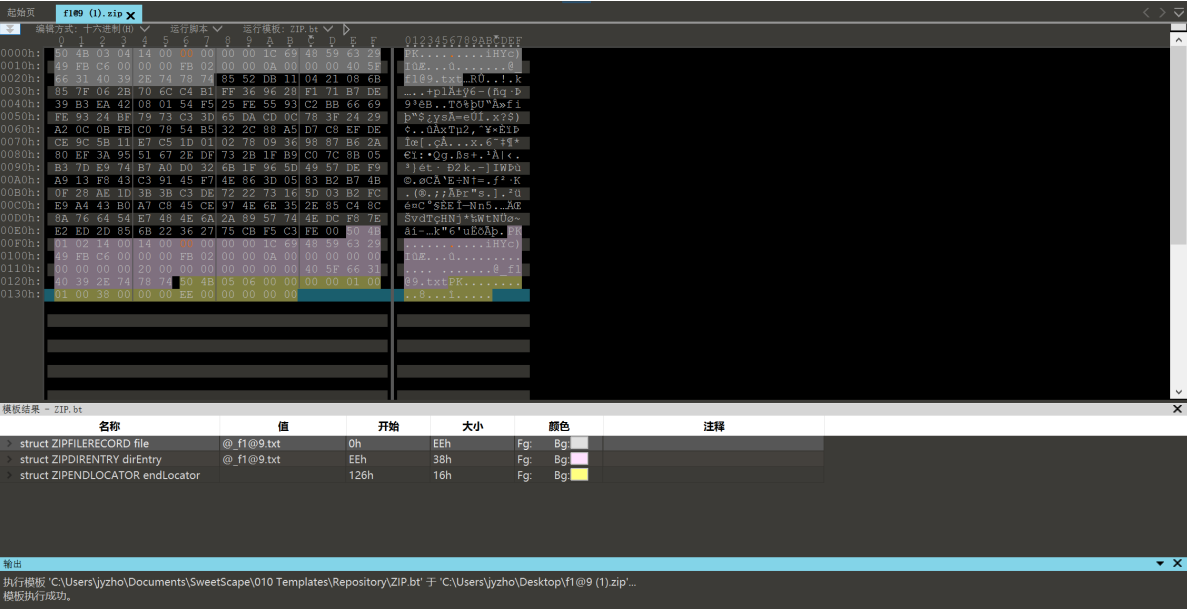
BuildCTF2024 Writeup

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

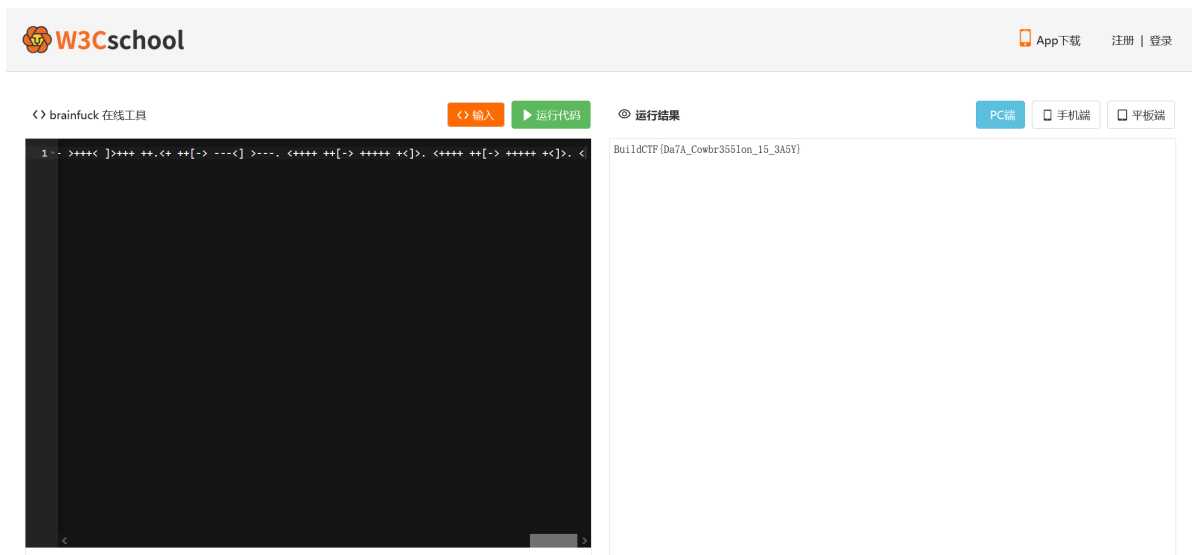


接下来按[这篇文章](#)做就行

```
(root@DESKTOP-LQMRD0K)-[/home/starr]
# printf '\x08' | dd of=1.zip bs=1 seek=$((0x08)) count=1 conv=notrunc
输入了 1+0 块记录
输出了 1+0 块记录
1 字节已复制, 0.000547584 s, 1.8 kB/s

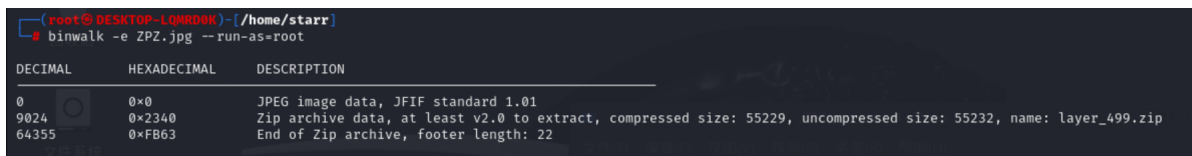
(root@DESKTOP-LQMRD0K)-[/home/starr]
# unzip 1.zip
Archive: 1.zip
inflating: @_f1@9.txt
```

txt的内容是brainfuck编码



EZ_ZIP

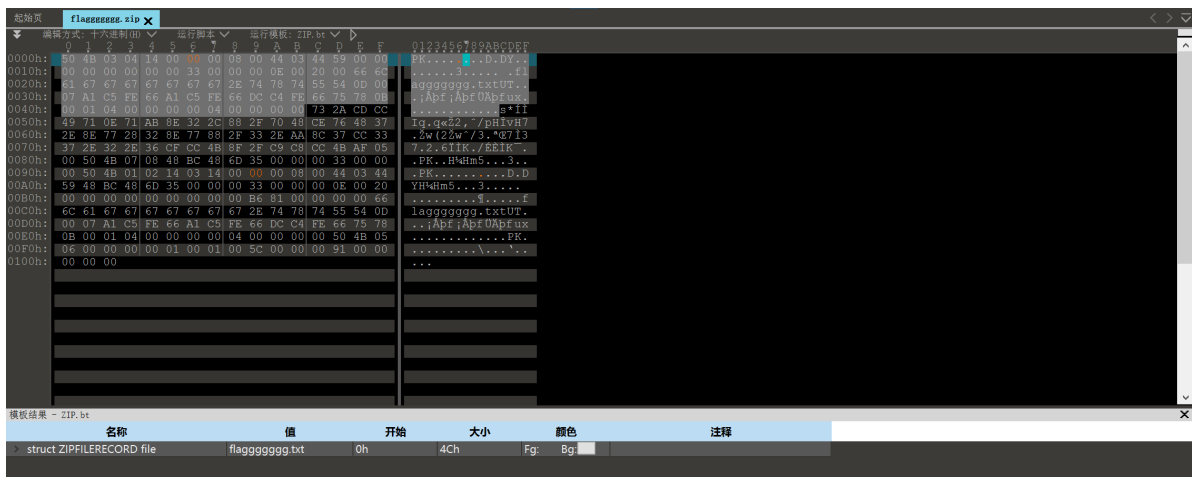
binwalk分离图片得到一个zip



看来是要解500层压缩包



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



HEX的秘密

如下

Recipe

From Hex

Delimiter

Auto

XOR Brute Force

Key length

1

Sample length

100

Sample offset

0

Scheme

Standard

☐ Null preserving
 ☒ Print key

☐ Output as hex

Crib (known plaintext string)

STEP

BAKE!

Auto Bake

Input

c2f5e9ece4c3d4c6bb3c5fafadfc1b5e3a1a1dfe2e9eeef2f9f9f9fd

Output

```

Key = 79: »...º.¿.Ê%...!¡.Øø!.....
Key = 7a: »...¹º%.Ê¿..»I.ÜÜ%.....
Key = 7b: »...¹%.Ê%..»øí.ÜÜ%.....
Key = 7c: %...¿"º.Í¹..Ê%É.ÝÝ%.....
Key = 7d: ¿...%º»..Í...¿%Ê.ÜÜ%.....
Key = 7e: %...%º..Í»...|¿É.ßß!.....
Key = 7f: %...%«².Íº..%Ê.ÞÞ.....
Key = 80: BuildCTF{3Ezz_A5c!!_binaryyyy}
Key = 81: CthmeBUGz2D{{^@4b ^cho`sxxy|
Key = 82: @wknfAVDy1Gxx|C7a##`'k1cp{{{.
Key = 83: Avjog@WExøFyy|B6`""`ajmbqzzz~
Key = 84: Fqmh`GPB.7A~~[E1g%[%fmjev}}y
Key = 85: GpliaFQC~6@..ZDøf$Zglkdw||x
Key = 86: DsojbER@}5C||YG3e`'Ydohgt...{
Key = 87: ErnkdSA|4B}}XF2d&&Xeni.fu~~z
Key = 88: Jjad1K\Ns;MrrWI=K)}Wjafizqqqu
Key = 89: K|`emJ]Or:LssVH<j((Vk`gh(pppt
Key = 90: H..cfaTø~80mVH3+Uhdqkssqv

```

what is this?

如下

Recipe

From Binary

Delimiter
Space

Byte Length
8

Substitute

Plaintext
c

Ciphertext
.

Substitute

Plaintext
p

Ciphertext
-

From Morse Code

Letter delimiter
Comma

Word delimiter
Line feed

Input

length: 1000
lines: 1

```
011000110110001101100011001011000111000001110000011100000111000001100000110000001011000110001101100011
0111000001110000011000110111000000101100011100000010110001100011011000110110001101100011000110001100
0110001101110000011100000111000001110000001011000110001101100011011000110011011000110001100011000110
01110000011100000110001101110000001011000110001101110000011100000111000000101100011000110001100011
01100011011000110110001101100011001011000110001101100011011100000111000001100011011000000101100
011100000111000000101100011100000111000001110000001011000110001101110000011000110111000001100011
001011000110001101100011011000110110001100101100011000110001100011000110001101110000
011100000110001101110000001011000111000001100011011100000110001100101100011100000111000001110000
0111000001110000001011000111000001100011011000110010110001100011001011100011000110110000
011100000110001101110000001011000111000001100011011100000110001101110000011100000010110001110000
0110001101110000011000110111000001110000
```

Output

time: 1ms
length: 24
lines: 1

SO_TH15_15_M0R5E_CODE_!!

一念愚即般若绝，一念智即般若生。

阴阳怪气编码，得到password

[illegible]

与佛论禅

CTF

编解码

编码转换

BigInt

字符处理

明文

MAC

对称加密(block)

对称加密(Stream)

非对称加密

签名与验证

Qrcode

PBE

Browser

Misc

ApiPost

压缩

图片模块

经纬度相关

关于

待处理:

📄

🔗

⋮

☐ 遍历

☒ 解密忽略空白符

佛曰：冥籍是响即俱藕蔴那在大姓词法多提潜涅冥咒孕大俱神闍佬老逝修依真响。卽呼帝冥耶梵大波得特輪至哆侑俱遠佐麻僧蘭真栗恐三夷罰醯伊羯俱若夷睡樂得鉢滅遙俱誅盡慧寫俱舍梵夜心語阿故哆上罰陀利得神佐帝响無端者俱曳眈轉仗夢伊道俱我曳電瞠樛老齡離冥實奢侏多修改悉但隸响阿詔一卽輪佐知任者機鑰至佳尼詔呼俱樂語夢鉢集壽離鉢。誨僧勝尼盡罰卑虛恐冥參多哆密苦怯畢即佐穰踐語屯詔至俱提罰帝俱槃梵恐俱轡上蒙陀罰勝無善數响俱所伽耶醢佐藐窺佐以楞詰明貴姪南心蠲倒佐夜遠故梵以楞囉世竟涅輸道奢倒冥穆罰得响摩持沙慧怯慧蠲但蠲槃鉢呼俱轉語明南沙寫竟槃上鉢故雖波無惡鉢漫孕羯佐慧利知暇疏哆蘇道响度知任响納得語婆明梵慧舍鉢謹奢竟俱數法能祛羅究歌夥夜帝羅

加密:

☐ 01248

☐ autoKey

☐ brain fuck

☐ DNA

☐ hacker words

☐ nihilist

☐ porta

☐ rot5

☐ troll script

☒ 与佛论禅

☐ 阴阳怪气

☐ a1z26

☐ bacon24

☐ bubbleBabble

☐ emojiSubstitute

☐ handyCode

☐ oneTimePad

☐ qwe

☐ rot8000

☐ twin-hex

☐ 六十四卦

☐ aaencode

☐ base64CaseCrack

☐ caesar

☐ Fenham

☐ hill

☐ Ook

☐ rabbit

☐ RSA-crack

☐ type7

☐ 兽音(online)

☐ ADFGX

☐ baudot

☐ caesar box

☐ fourSquare

☐ jjenode

☐ pawnShop

☐ railFence

☐ socialistCoreValue

☐ virgenene

☐ 天干地支(base60)

☐ affine

☐ beaufort

☐ cetacean

☐ fracMorse

☐ manchester

☐ periodicTable

☐ rot13

☐ steg base64

☐ vowel

☐ 新佛日(online)

☐ asciiSum

☐ Citrix CTX1

☐ grayCode

☐ manchester-diff

☐ playFair

☐ rot18

☐ tapCode

☐ zwBinary

☐ 熊日(online)

☐ atbash

☐ braille(盲文)

☐ curveCipher

☐ gronsfeld

☐ morse

☐ polybius

☐ rot47

☐ trifold

☐ zwUnicode

☐ 百家姓

☐ encrypt version

custom password for encrypt version, default is TakuronDotTop

☐ 加密

☒ 解密

▶ 运行

quippuiq.com

wiki

输出内容:

📄

↑

日：坤元元量华幼始纳灵+梵冥杰渺荡净+浩虚玉道坤玉终罗杰度清魄魔神龙融魂玉命魄电照色玉冥周色西鬼终命东融量照霄杰北南西北生配+空茫陀周度清道茫杰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: BuddhaSav 输入长度: 310 输出长度: 100 cost: 5 ms

天书解码

密文↓(字:100)

密钥key/str/url:

曰：坤芒元里华劫始南灵+梵冥无渺荡净+浩虚玉道坤王终罗无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青杰北南西生陀+空芒陀周度清道芒杰冥度东真阎阎阿度命净威芒度元芒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↑

百家姓磁贴解码:

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:

盲文解码: 曰：坤芒元里华劫始南灵+梵冥无渺荡净+浩虚玉道坤王终罗无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青杰北南西生陀+空芒陀周度清道芒杰冥度东真阎阎阿度命净威芒度元芒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：坤芒元里华劫始南灵+梵冥无渺荡净+浩虚玉道坤王终罗无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青杰北南西生陀+空芒陀周度清道芒杰冥度东真阎阎阿度命净威芒度元芒微威陀空人罗北魂威荡幽吉灵尊

火星文解密2: 曰：坤芒元里华劫始南灵+梵冥无渺荡净+浩虚玉道坤王终罗无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青杰北南西生陀+空芒陀周度清道芒杰冥度东真阎阎阿度命净威芒度元芒微威陀空人罗北魂威荡幽吉灵尊

棋盘密码解码:

八卦符号解码:

中文域名PunyCode解码:

键盘包围解码:

键盘按键键值(keyCode)解码:

键盘上档键解码: 曰：坤芒元里华劫始南灵+梵冥无渺荡净+浩虚玉道坤王终罗无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青杰北南西生陀+空芒陀周度清道芒杰冥度东真阎阎阿度命净威芒度元芒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

密文↓(字:48)

密钥key/str/url:

一键解码

7dcUypxFHgNSJ7rVkdM5k7X18GSU9JinEziQ5vsS1rMpvTpx↓

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↑

一键解码:

|解密结果(注:在线解密密码不参与一键解码)

base64解码: mW□JE□R':U□S9□5upd□+□'□8□f(□V3)=:q

base32解码:

base16解码:

base85(a)解码:

base85(b)解码:

base58解码: BuildCTF{D3crypt10n_1s_4_10ng_r04d}

base36解码:

base91解码:

base92解码: =Å%ŷŁ&ŷa?U&MIİ&×3[4□0&ŷ

'-fFDN)E□0>[ŷ{

base62解码: 13348996933733200377760866035747250623921089262095567381252110270620050246861207929457

base62(ASCII)解码:

Base16-32-64-91混合多重解码:

1 isBase64 True 7dcUypxFHgNSJ7rVkdM5k7X18GSU9JinEziQ5vsS1rMpvTpx

2 解码结果: b"\xed\xed\x14\xca\x9cE\x1e\x03R'\xba\x5\x91\xd39\x93\xb5\xf5\xf0d\x94\xf4\x98\xa7\x138\x9

0\xe6\xf6\x12\xd6\xb3)\xbd:q

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:

摩斯解码:

埃图解码:

7m1Y1nt+00mFW&7:7hmD5h7T18PWVQ&reMrxH5zmW1:DerVat

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.101o.xyz

四妹, 你听我解释

扔进010, 看到结尾多了点东西

010 Editor - C:\Users\lyzho\Desktop\c.png

文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(M) 模板(T) 调试(D) 工具(I) 窗口(W) 帮助(H)

起始页 c.png 十六进制(H) 运行脚本 运行模板: PNG.bt

7:5690h: EF CE E6 B6 7E AE E6 9D 9E 45 56 A5 03 A4 13 7123456789ABCDEF
7:5660h: E3 0B 94 3D 51 EF 59 5D 0D A4 02 87 76 04 01123456789ABCDEF
7:5670h: 36 9C 1F 1C 0D 78 00 7F 79 2F 88 75 8D E4 09 3D 023456789ABCDEF
7:5680h: 77 C2 1B 68 C5 5F 1E 19 A5 AF 3F EC E7 EF CE 57 03456789ABCDEF
7:5690h: AF 5F B1 02 F2 56 65 C8 5D AF 8D BE 70 F2 E7 71 0456789ABCDEF
7:56A0h: 6F 1E AA 1A C9 18 63 81 AF 02 DF 3E 64 F5 13 47 056789ABCDEF
7:56B0h: E4 0B 36 7D D9 58 4E 00 EE 1D AC BE E3 EF 07 6E 06789ABCDEF
7:56C0h: D4 F3 F1 AA 35 09 00 00 00 00 47 25 1E 44 AE 0789ABCDEF
7:56D0h: 0A E6 E7 E0 87 AA E7 94 B1 E6 96 87 E6 98 8E E6 089ABCDEF
7:56E0h: B3 95 E6 B2 BB E5 B9 B3 E7 AD B9 E5 85 AC E6 AD 09ABCDEF
7:56F0h: A3 E6 95 AC E4 B8 9A E3 95 AC E6 AD A3 E5 B8 BB 10ABCDEF
7:5700h: E5 96 84 E5 85 AC E6 AD A3 E5 85 AC E6 AD A3 E8 11ABCDEF
7:5710h: 87 AA E7 94 B1 E8 87 AA E7 94 B1 E5 92 8C E8 B0 12ABCDEF
7:5720h: 90 E5 B9 B3 E7 AD B9 E8 87 AA E7 94 B1 E8 87 AA 13ABCDEF
7:5730h: E7 94 B1 E5 85 AC E6 AD A3 E6 B3 95 E6 B2 BB E5 14ABCDEF
7:5740h: 6F 8D E5 96 84 E5 B9 B3 E7 AD B9 E5 85 AC E6 AD 15ABCDEF
7:5750h: A3 E5 AF 9A E4 BF A1 E6 96 87 E6 98 8E E5 85 AC 16ABCDEF
7:5760h: E6 AD A3 E6 B0 91 E4 B8 BB E5 85 AC E6 AD A3 E8 17ABCDEF
7:5770h: AF 9A E4 BF A1 E5 B9 B3 E7 AD B9 E5 B9 B3 E7 AD 18ABCDEF
7:5780h: 89 E9 AF 9A E4 BF A1 E5 B9 B3 E7 AD B9 E6 B3 95 19ABCDEF
7:5790h: E6 B2 BB 20ABCDEF

模板结果 - PNG.bt

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, Public, ...	21h	Dh	Fg: Bg:	
struct PNG_CHUNK chunk[2]	gAMA (Ancillary, Public, ...	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:	
struct PNG_CHUNK chunk[6]	IDAT (Critical, Public, ...	30004h	10000h	Fg: Bg:	

输出

你打开的 C:\Users\lyzho\Documents\SweetScape\010 Templates\Repository\PNG.bt 于 C:\Users\lyzho\Desktop\c.png ...

ERROR: CRC Mismatch @ chunk[0]: in data: c5a18d9c; expected: d073b961

ERROR Line 332: 声明中的数组大小无效。

复制出来from hex, 得到前半串编码

Recipe

From Hex

Delimiter
Auto

Input

length: 576
lines: 13

E8 87 AA E7 94 B1 E6 96 87 E6 98 8E E6 B3 95 E6
B2 BB E5 B9 B3 E7 AD 89 E5 85 AC E6 AD A3 E6 95
AC E4 B8 9A E5 85 AC E6 AD A3 E5 8F 88 E5 96 84
E5 85 AC E6 AD A3 E5 85 AC E6 AD A3 E8 87 AA E7
94 B1 E8 87 AA E7 94 B1 E5 92 8C E8 B0 90 E5 B9
B3 E7 AD 89 E8 87 AA E7 94 B1 E8 87 AA E7 94 B1
E5 85 AC E6 AD A3 E6 B3 95 E6 B2 BB E5 8F 88 E5
96 84 E5 B9 B3 E7 AD 89 E5 85 AC E6 AD A3 E8 AF
9A E4 BF A1 E6 96 87 E6 98 8E E5 85 AC E6 AD A3
E6 B0 91 E4 B8 BB E5 85 AC E6 AD A3 E8 AF 9A E4
BF A1 E5 B9 B3 E7 AD 89 E5 B9 B3 E7 AD 89 E8 AF
9A E4 BF A1 E5 B9 B3 E7 AD 89 E6 B3 95 E6 B2 BB

Output

time: 1ms
length: 64
lines: 1

自由文明法治平等公正敬业公正友善公正公正自由自由和谐平等自由自由公正法治友善平等公正诚信文明公正民主公正诚信平等平等诚信平等法治

同时还报crc有问题, 猜测需要爆破宽高

```
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

—— 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

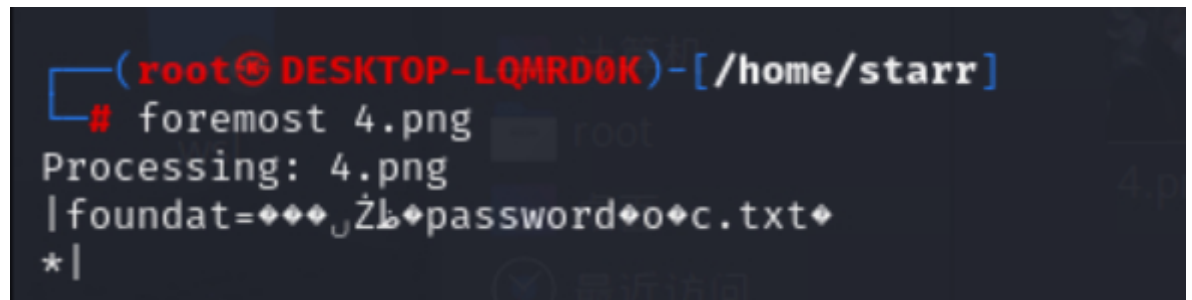
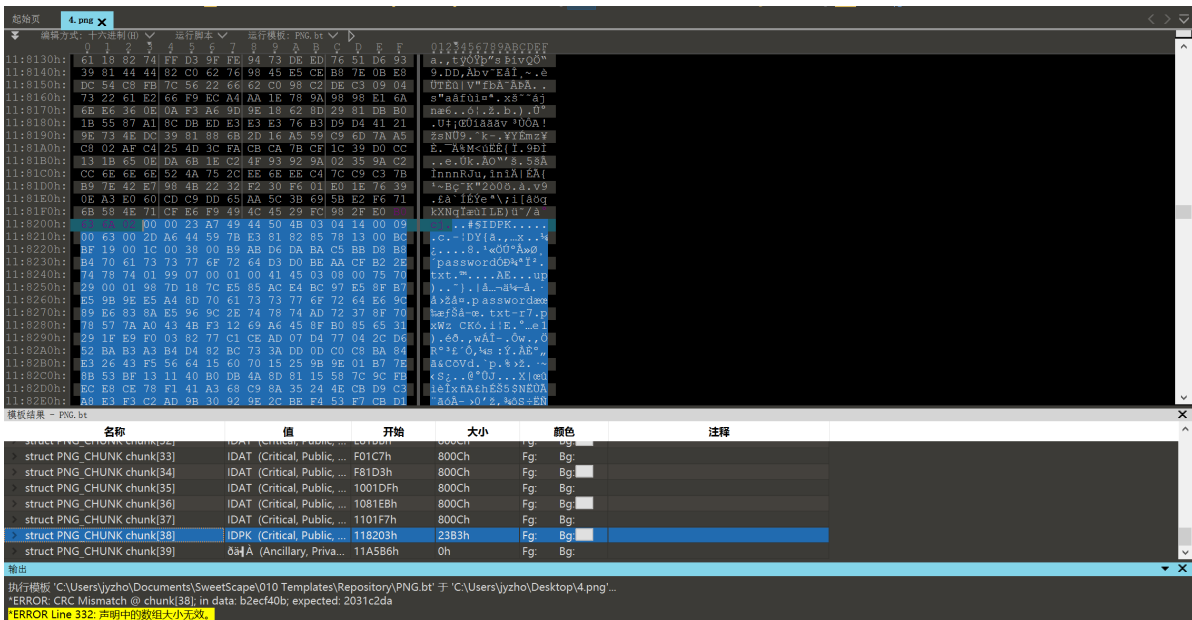
```
BuildCTF {lao_se_pl}|
```

编 码 解 码

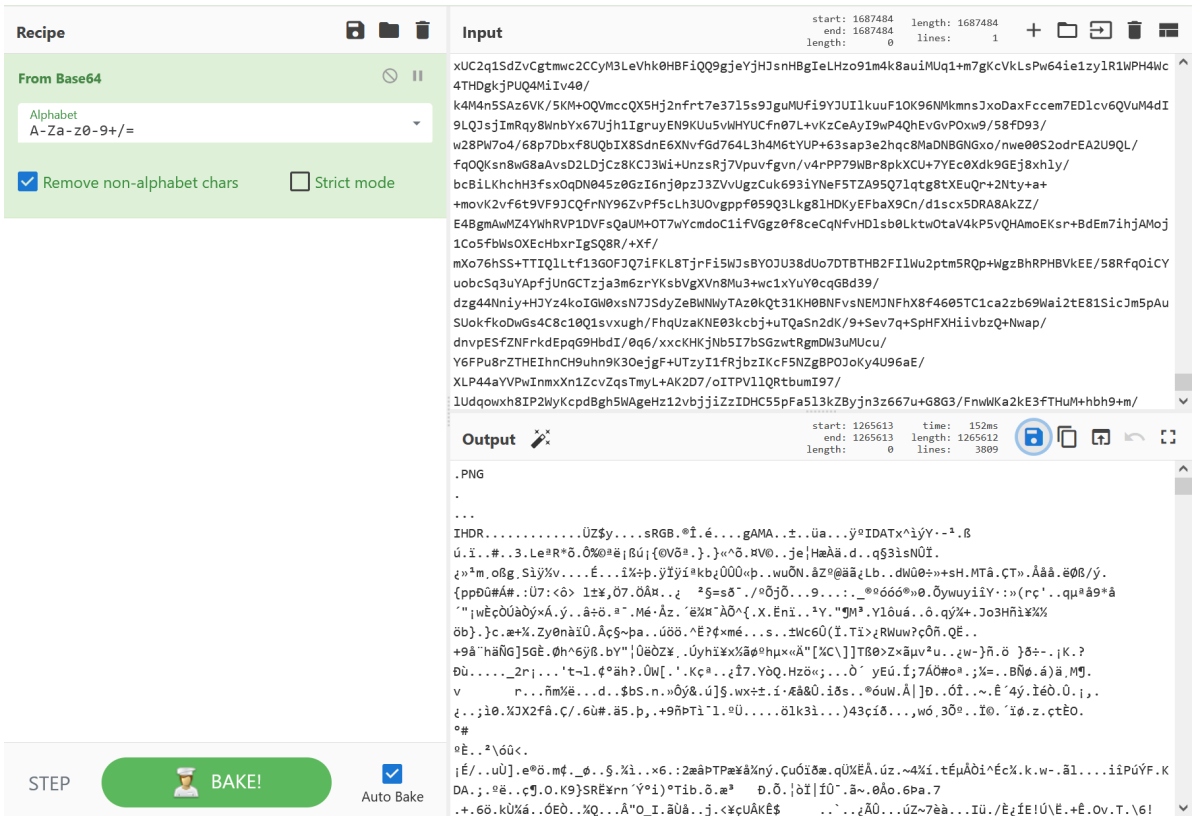
自由文明法治平等公正敬业公正友善公正公正自由自由和谐平等自由自由公正法治友善平等公正诚信文明公正民主公正诚信平等平等诚信平等法治和谐公正平等平等友善敬业法治富强和谐民主法治诚信和谐

四妹？还是萍萍呢？

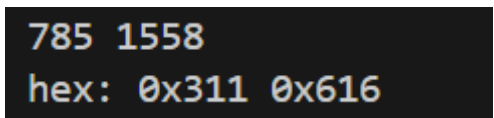
在图片的第38个chunk处发现了压缩包的部分，加个504B上去丢给foremost去分解



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。

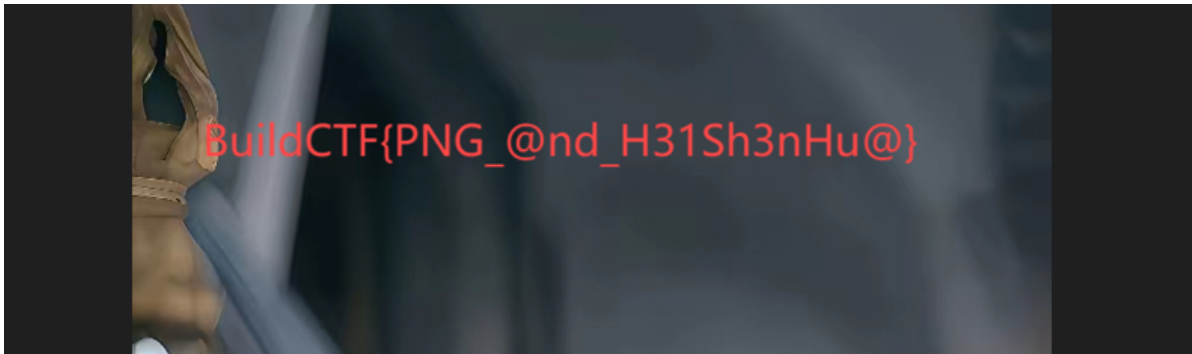


图片的状态一眼就是宽高有问题，去爆破一下，然后修改，得到flag。



Hex editor view of a PNG file. The hex data is shown in columns, with the corresponding ASCII characters on the right. The PNG signature is visible at the top.

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, Public, ...)	21h	Dh	Fg: Bg:	
struct PNG_CHUNK chunk[2]	gAMA (Ancillary, Public, ...)	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:	
struct PNG_CHUNK chunk[6]	IDAT (Critical, Public, ...)	30004h	10000h	Fg: Bg:	

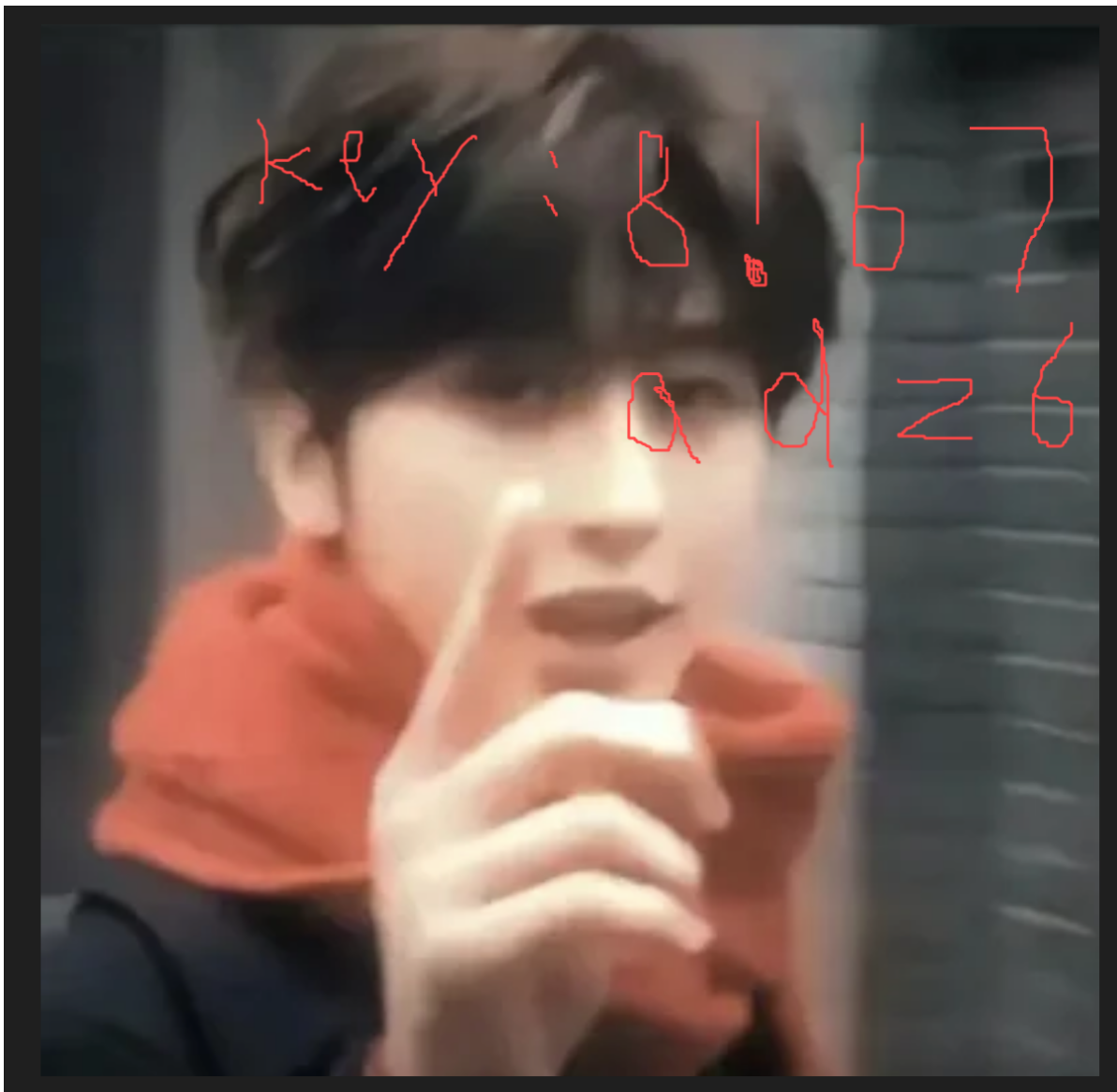


如果再来一次，还会选择我吗？

可以看出每两个字节交换了一下，写脚本还原，得到key

Hex editor view of a file named 'password.png'. The hex data is shown in columns, with the corresponding ASCII characters on the right. The PNG signature is visible at the top.

```
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')

```

在线条码阅读器

上传图片，选择条形码类型或留下“所有类型”，然后点击“阅读条形码”按钮。
由 aspose.com 和 aspose.cloud 提供支持

另一张图片



键入: Code 128

key:wo_bu_shi_xiao_hei_zi!!!

生成新的

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

Recipe

sav folcdel

From Base64

not interested pause

Alphabet

A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

From Base64

not interested pause

Alphabet

A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

From Base64

not interested pause

Alphabet

A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

From Base64

not interested pause

Alphabet

A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

From Base64

not interested pause

Alphabet

A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

STEP

CHEF ICON BAKE!

☒ Auto Bake

Input

start: 162536 end: 162536 length: 162536

length: 0 lines: 1

adofolkinp delvie

Vm0wd2QyUX1VW6xwV0d4V1YwZDRMMV13WkRSV01WbDNXa1JTVjAxV2J2ET1hhMUpUVmpBeFyYsKvUbGhoTVVwVZtcE1R115U2twVWJHaG9UV1Z3V1ZadGNFSmxSbGw1VTJ0V1ZX5khhRz1VVMxaM1ZslwFkR05GU214U2JHdzFWVEowVjFawFNraGhS emxwVm14YU0xWnNXbUZrUjA1R1UyMTRVMkpIZHpGV1ZFb3dWakZhV0Z0cmFHaFNlBhXVWm8NF1VMHhXbk5YY1Vac1VqQT FSMV5TVRSVk1rcE1aSHBVjFaRmIzZFdha1poVjBaT2NtRkhhRk5sY1hoWFZtHhORmxWTUhoWjGjrNV1ZbFZhY2xwVcVU R1NNV1Y1VFZSU1ZrMXJjRwXhU0hCSFZqRmFSbU16WkZkaGExcG9WakJhVDJ0dFJraGhSazVzWxob1dGwNRNSGhPUm14V1 RVaG9XR0pyT1ZslwJGwMhZMnhXY1ZGVVJsTk5WbFkxVkaU1UxWnJNWepqU1d4aFUwaENTR1pxUm1GU2JvBdZxa1prYUdF eGNHOvdha0poVkrKT2RGSnJhR2hTYXpWe1dXeG9iMwRHV25ST1NHafBVbTE0VjF5VmfH0VhSMHB5VGxac1dtSkdXbWhaTW 5oWfKxwkdWVkpzVGs1V2JGa3hWa1phVTFVeFduSk5XRXBxVWxkNGF6VXdhRUSUUmXweFVtMudVMkpWYkRaWGEcxHJZVMRG ZUD0SE9WZGhHMHbVmtSS1QyUkdTbkpoUjJoVf1YcFd1bGRYZuc5aU1XUkhMjJVTGxOSGFGQ1ZiVEUwmpGU1ZtRkKpVm hTTUhcCNVZHeGFjMwR0U2tkWGXaGFUVzVvV0ZreFdrZFdwa3B6VkdzMVYySkdhM2hXYTFwaFZUR1Z1RmR1U2s1WFJYQnhW V3hrTkDGR1ZYZGhSVTVVW14d2VGvNRNVWRWTwtwV11rUmFXR0V4Y0hKW1ZXUkdaVWRPU0U5V1pHaGhNSEJ2Vm10U1MxUX 1Va2RUYmtwv1VqSm9WRmxZY0ZkbGJH11aVNM1VYUxWfVraFdNa1ZUVkd4T1NHRkdRfppVkvVd1ZtcEdVMvp0Umt0UFZt aFRUvWHD1ZaSGVRHmpNv1IwVTJ0a1dHSLhhR0ZVWnpld1YwWnJ1RmRyWkZKv2EzQjZwa2R6TVZzeVNRzGhNMhYwVRGd2 FGIW1VSbFpsUm1SMVUyczZFXRkpZUM5oV1YzaHJua2RHUjFawVpHaFNwVFWV1cxNGQyVkdwb1JOV1dSV1RXdHdWmXyVw1G WFIwVjRZMGhLV2xawFVrZGfWV1JQVTBVNVYxcEhhR2hOU0VKM1ZtMTBVMU14VhVsVmEyU1VZbXRY3YjFwcVNT0vdsBxhawT Baa2JH5KhVbGxhV1dNMV1VWVhXRL1ZYUZkTmFsw1VWa2Q0VDF0SFJrZFJiRnBwVmtkVwMqXwNRjRMRWTVZwMFVtdG9VR1p0 YUZSVVZxaERUbFphU0dWsfJtce5WMMU13V1RKMGEZehTbGhoUjBaV1ZucFdkbF13V25KbFJtUn1Xa1prVjJFe1FqW1dhM1 I2VFZawmVWtnJaR2h0TW10wVdWUkdMkZHV2xWU2JGcHNvTFTTtVZVeWN6R1hSa3BavVc1b1YxWxPhSEpVYTJSSFFvQrmFX

Output

start: 30 end: 27 length: -3

time: 58ms length: 27 lines: 1

sav coropeunc full

BuildCTF{y0u_are_great_boy}



lsb

Extract Preview

516e567062475244	56455a374d563968	QnVpbGRD VEZ7MV9h
625639754d485266	54464e4349534568	bV9uMHRf TFNCISEh
66513d3d00000000	0000000000000000	fQ==
0000000000000000	0000000000000000	
0000000000000000	0000820000124900	
28beda87e3841043	f0bfbeffbfd7428	(.....Ct(
0000924924905c7b	fdfedeba49248000	...I\$. \{ ...I\$. ..
0000492495a3a142	40301f6fabd04800	..I\$. ...B @0.o..H.
004826180d2fe81d	243df7dbd04055c0	.H&.../. \$=...@U.
e381ea5d00ba5ea1	ed48549249249240	...]. ..^.. HT.I\$. @

Bit Planes

Alpha ☐ 7 ☐ 6 ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1 ☐ 0

Red ☐ 7 ☐ 6 ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1 ☒ 0

Green ☐ 7 ☐ 6 ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1 ☒ 0

Blue ☐ 7 ☐ 6 ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1 ☒ 0

Preview Settings

Include Hex Dump In Preview ☒

Order settings

Extract By ☒ Row ☐ Column

Bit Order ☒ MSB First ☐ LSB First

Bit Plane Order

☒ RGB ☐ GRB

☐ RBG ☐ BRG

☐ GBR ☐ BGR

Preview Save Text Save Bin Cancel

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars ☐ Strict mode

Input

QnVpbGRDVEZ7MV9hbV9uMHRfTFNCISEhfQ==

Output

BuildCTF{1_am_n0t_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 =
82798533307572346691364830327508248261757779275065750837101664128970120794669557
69715275604152872242147320194640165649152928984919315754419447729793483984130396
35857857113795657130251620264907661907683199792267557270584819950430923204450295
78663170112125059852841293655225703683953684273889042237827428506169831308851527
85650513046301920305069822348366931825404271695876688539675285303882189060671184
91113974255471001875556551801477773332279552271023409135387829848649824482963887
89493896903844885733381388256423816877498881023413792541374455463067962580927620
99409409285871651688611387507673794784257901946892698481

# Q =
94066435031767666881139042267024773227066647317142726325257635333953802983201403
41860043591350428258361089106233876240175767826293976534568274153276542755524620
13871476733882033474814036508085647425333403323645709276424499498383791495528680
81537846287393272175397011349397483131230713476978272791699528107279956817807177
19971161661561936180553161888359929479143712061627854343656949334882218260141557
76886822215146847194688422537000970690064085149279853845838444929404293083135972
37998935815686774338685316993607898004490777517985354971170040597346709128293587
93175346866262442550715622833013235677926312075950550681

# n =
88252834821904760059462533436388208795593553068609122681288912415133100540664245
67824202757539757712177309282694997613217968336164050770152277369601415394249781
57741545622412010254396828503564751446136461173433807352345435437699278355103539
55581941712026808551828687665632776973256902268493169441017394916598121740548985
19492145495098671439125714086449826697343692081109131564556220174583970363431110
46222247301302182577026780324951589373698943014619419993633515393661119646722559
97468308739580852876652231907671374043668400552978595544901233898773969657101772
79558954630222879974581602069901175074777191362537419581

# e = 65537
```



```
# c =  
27915082942179758159664000908789091022294710566838766903802097394437507062054409  
03393230396682009623237564687348042748584473338129846717106998541823787312098413  
21663432583453894778443392614883185887601252309793406780068717541254872792121209  
45061845738130108370814509280317816067243605608952074687396728904772649873860508  
24080954154593921962425487890029112673939096782014103626071220855557452213144655  
65955623309692096652917573862466480609908407877697721605498625381163709053064022  
93764494501838709895355570646716245976733542014165663539815972755562821443411642  
64798189863676182210722120396629675835054747757641121674459453400205767362567818  
88244765432880489561245655094731005508385630855854346757273588316107249205502133  
50035792170323729397796947598697983084347567191009236345815968927729025919066227  
70472818006080555378715186242603427552660515480784069549864407018468196231163933  
82734698598385053488234172347222707988823843670586300641081552406803077545574724  
76430983184039474907188578578484589833812196216551783354411797156409948499012005  
96394372856480389815015573576269582565867847574655990070579681451283838019360317  
86572260334068128103149601422510122235769841156423514636847245124567785488530026  
53596485899854303126091917273560
```

简单的数学运算，不多说了

```

import gmpy2
from Crypto.Util.number import *
P =
8279853330757234669136483032750824826175777927506575083710166412897012079466955
76971527560415287224214732019464016564915292898491931575441944772979348398413039
63585785711379565713025162026490766190768319979226755727058481995043092320445029
57866317011212505985284129365522570368395368427388904223782742850616983130885152
78565051304630192030506982234836693182540427169587668853967528530388218906067118
49111397425547100187555655180147777333227955227102340913538782984864982448296388
78949389690384488573338138825642381687749888102341379254137445546306796258092762
099409409285871651688611387507673794784257901946892698481
Q =
9406643503176766688113904226702477322706664731714272632525763533395380298320140
34186004359135042825836108910623387624017576782629397653456827415327654275552462
01387147673388203347481403650808564742533340332364570927642449949838379149552868
08153784628739327217539701134939748313123071347697827279169952810727995681780717
71997116166156193618055316188835992947914371206162785434365694933488221826014155
77688682221514684719468842253700097069006408514927985384583844492940429308313597
23799893581568677433868531699360789800449077751798535497117004059734670912829358
793175346866262442550715622833013235677926312075950550681
n =
8825283482190476005946253343638820879559355306860912268128891241513310054066424
56782420275753975771217730928269499761321796833616405077015227736960141539424978
15774154562241201025439682850356475144613646117343380735234543543769927835510353
95558194171202680855182868766563277697325690226849316944101739491659812174054898
51949214549509867143912571408644982669734369208110913156455622017458397036343111
04622224730130218257702678032495158937369894301461941999363351539366111964672255
99746830873958085287665223190767137404366840055297859554490123389877396965710177
279558954630222879974581602069901175074777191362537419581
e = 65537
c =
2791508294217975815966400090878909102229471056683876690380209739443750706205440
90339323039668200962323756468734804274858447333812984671710699854182378731209841
32166343258345389477844339261488318588760125230979340678006871754125487279212120
94506184573813010837081450928031781606724360560895207468739672890477264987386050
82408095415459392196242548789002911267393909678201410362607122085555745221314465
56595562330969209665291757386246648060990840787769772160549862538116370905306402
29376449450183870989535557064671624597673354201416566353981597275556282144341164
26479818986367618221072212039662967583505474775764112167445945340020576736256781
88824476543288048956124565509473100550838563085585434675727358831610724920550213
35003579217032372939779694759869798308434756719100923634581596892772902591906622
7704728180060805537871518624260342755266051548078406954986440701846819623116393
38273469859838505348823417234722270798882384367058630064108155240680307754557472
47643098318403947490718857857848458983381219621655178335441179715640994849901200
59639437285648038981501557357626958256586784757465599007057968145128383801936031
78657226033406812810314960142251012223576984115642351463684724512456778548853002
653596485899854303126091917273560
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
412217389320600888361578917153088073678587422269955710471
n2 =
11933661747067216317642315621042074566046499785197709817779978157416906347669444
37423431332906485962296074374351173567261499956626402564869858988618505675807171
83199642626198191437579229166241963543133224565342665201505430081178881013499203
9673753293761650268966720820732904897987222563877933742673021891249520999021187
40406570638870071120844562804138695645939827123023601847696483939924514366653435
91137778465351517731747017322842800835865804899956663063738394179466481961408799
78268472361473557375951972193618245984950374326806423407152520541682571610372434
453778172497925696535270204943842467472100237854318244291
```

```

c =
20080676122944896238797522372441559951736929534371084097400233944319893926800196
69444956453415077008555434995243314181563732475338648454961657363600176381585209
59848308289520200479384069092743117853062990610216624845443718137397135203613433
50959698642021322243662988875917088108399877176033404097457939417134483333264562
6026338536943820144727475001591007236263149284764840376651985760456830096707186
81515081427842710426008154068539786968573097609511058522883546035032073838999021
35741426285551161292195639862478256231538619968275273876467583013024899054710124
331145912185471501398910765579441956531091561893256832468
h0 =
29967260097262606957328211665040403447311026370476824328840588574939356250942580
46641569918904978173116793673563730117949606727933902262668880339210084101176866
38360254396617984035363373550744292670734225839136224590485029741664227112332898
08129310256778573731995401292800973158329070237770521011336498771944954805436464
72133854655383755313968952550827443970931104462445312146328606862802196901953935
23897275985243588272078657096554228627854910740291804119400884571750773578689796
87348310643933377735578178393434490013685658569211384080399316088042335959804975
57733714560035682416265029819340316734845279080134432704
h1 =
19843160604742228074331688651361052208481287636527838615063387670722213224954610
44872006593737820154517727884157563369701243407418604655684329206883575211338475
61499441142989491154128197308435982886372594670852688612017757238177904283865955
59040938133481222229290199923979132846871398172318539492741755408720073350962388
13845334167700954761623826221117672742406794602068374226278231973528635746581778
64462385281877229593574446765127054515041363333364158800205025240096479401827212
64953084120705872870651891290569527156804993340563927419561415555818468261824287
933683736509372616293569615247228388443284457740072850735
h2 =
15147052684674827267989051566164167603473413362261253296001082161136918959833294
46318533541666212736847398023966791856160074166751328570884308147507468823950733
02305583314088775832466618620409184100369365053074373299143632016302121639523574
44441705663871720438955166472073576526814546767805314463827075388036712200327696
16896576217756734696647939989657819011181913000099159449093238813218824172665475
63686989982328263409692880826458603244049801434894899464902664394473424614834905
82149239131554246756547000945718737195930407251232848166108751122870333559461452
459416252942341423373918245090162970624108991537972775066
"""

```

不多说了，丢个[链接](#)在这吧

```

from Crypto.Util.number import *
from gmpy2 import *
n1 =
19957426023169626195602761840035904096149402534966487535713447987366768645542881
12478255126897834206345843084687782421065977812628170598471106119035163649794494
33219889501881711599037173489365563461986383119500161368654250150370982700400318
72702873264144372191898253134939805153141701819590164140250130420280491966786900
65118694131795955606673095974427996397606556543615339967947541004077363714267793
69268946779192423516104572962038648069915394805935460844493230176704315900123125
26757477514457145686070196978477495658962519391041011847512041022828710693830661
412217389320600888361578917153088073678587422269955710471

```

```

n2 =
11933661747067216317642315621042074566046499785197709817779978157416906347669444
37423431332906485962296074374351173567261499956626402564869858988618505675807171
83199642626198191437579229166241963543133224565342665201505430081178881013499203
96737532937616502689667208207329048979872222563877933742673021891249520999021187
40406570638870071120844562804138695645939827123023601847696483939924514366653435
91137778465351517731747017322842800835865804899956663063738394179466481961408799
78268472361473557375951972193618245984950374326806423407152520541682571610372434
453778172497925696535270204943842467472100237854318244291

c =
20080676122944896238797522372441559951736929534371084097400233944319893926800196
69444956453415077008555434995243314181563732475338648454961657363600176381585209
59848308289520200479384069092743117853062990610216624845443718137397135203613433
50959698642021322243662988875917088108399877176033404097457939417134483333264562
60263385369438201447274750015910072362631492847648403766651985760456830096707186
81515081427842710426008154068539786968573097609511058522883546035032073838999021
35741426285551161292195639862478256231538619968275273876467583013024899054710124
331145912185471501398910765579441956531091561893256832468

h0 =
29967260097262606957328211665040403447311026370476824328840588574939356250942580
46641569918904978173116793673563730117949606727933902262668880339210084101176866
38360254396617984035363373550744292670734225839136224590485029741664227112332898
08129310256778573731995401292800973158329070237770521011336498771944954805436464
72133854655383755313968952550827443970931104462445312146328606862802196901953935
23897275985243588272078657096554228627854910740291804119400884571750773578689796
87348310643933377735578178393434490013685658569211384080399316088042335959804975
57733714560035682416265029819340316734845279080134432704

h1 =
19843160604742228074331688651361052208481287636527838615063387670722213224954610
44872006593737820154517727884157563369701243407418604655684329206883575211338475
61499441142989491154128197308435982886372594670852688612017757238177904283865955
59040938133481222229290199923979132846871398172318539492741755408720073350962388
13845334167700954761623826221117672742406794602068374226278231973528635746581778
64462385281877229593574446765127054515041363333364158800205025240096479401827212
64953084120705872870651891290569527156804993340563927419561415555818468261824287
933683736509372616293569615247228388443284457740072850735

h2 =
15147052684674827267989051566164167603473413362261253296001082161136918959833294
46318533541666212736847398023966791856160074166751328570884308147507468823950733
02305583314088775832466618620409184100369365053074373299143632016302121639523574
44441705663871720438955166472073576526814546767805314463827075388036712200327696
16896576217756734696647939989657819011181913000099159449093238813218824172665475
63686989982328263409692880826458603244049801434894899464902664394473424614834905
82149239131554246756547000945718737195930407251232848166108751122870333559461452
459416252942341423373918245090162970624108991537972775066

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)

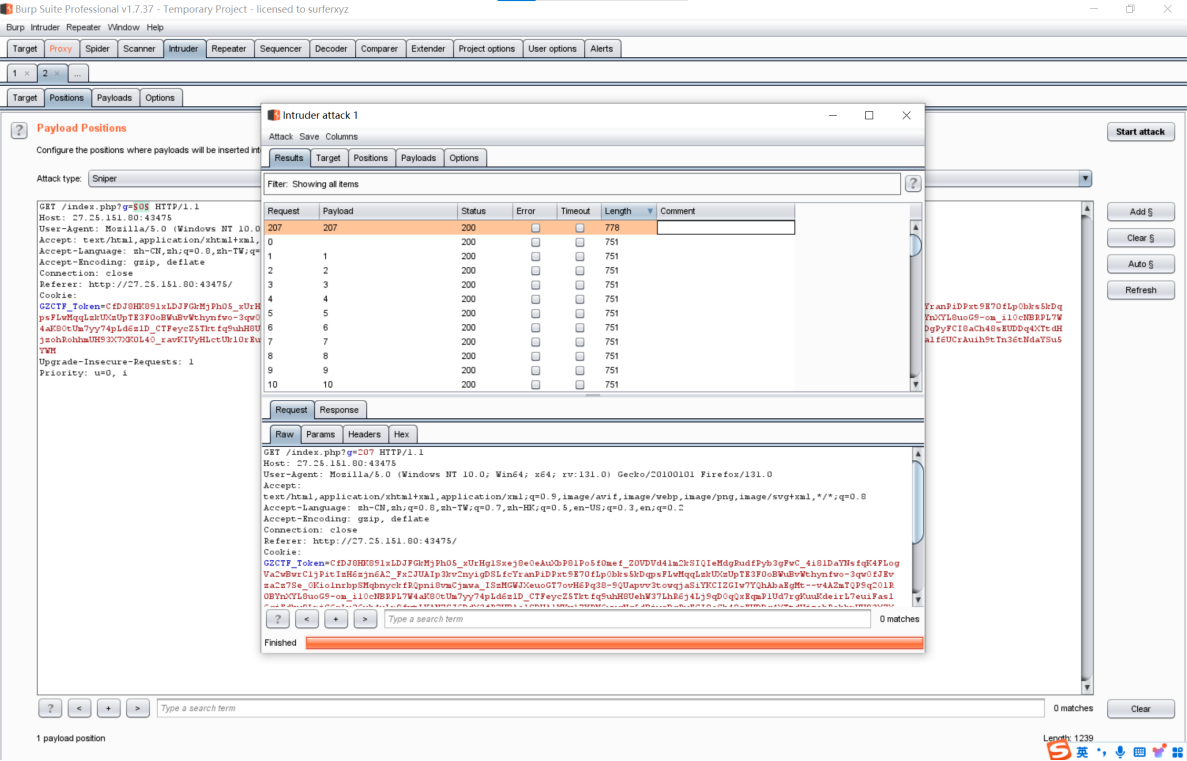
phi=(p-1)*(q-1)
d=invert(e,phi)
m=pow(c,d,p*q)
print(long_to_bytes(m))

```

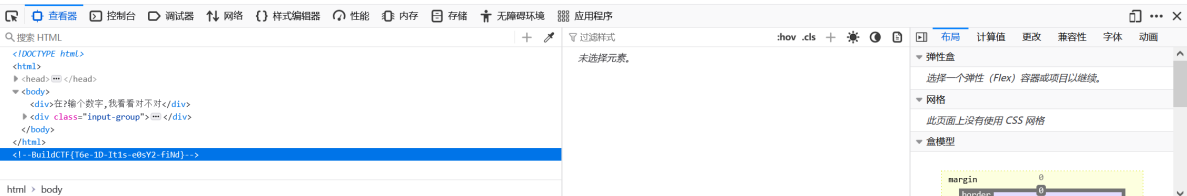
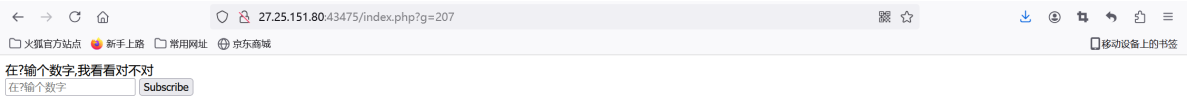
Web

find-the-id

用bp爆破一下可以看出应该是207

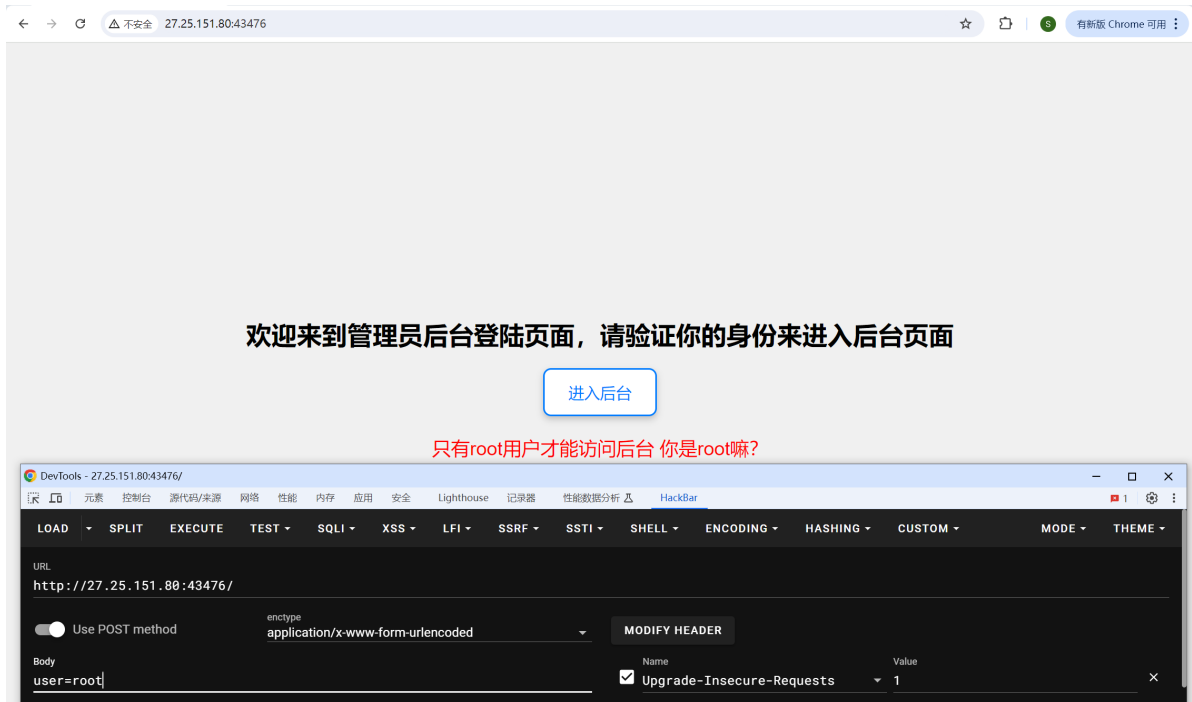


可以在前端注释中找到flag

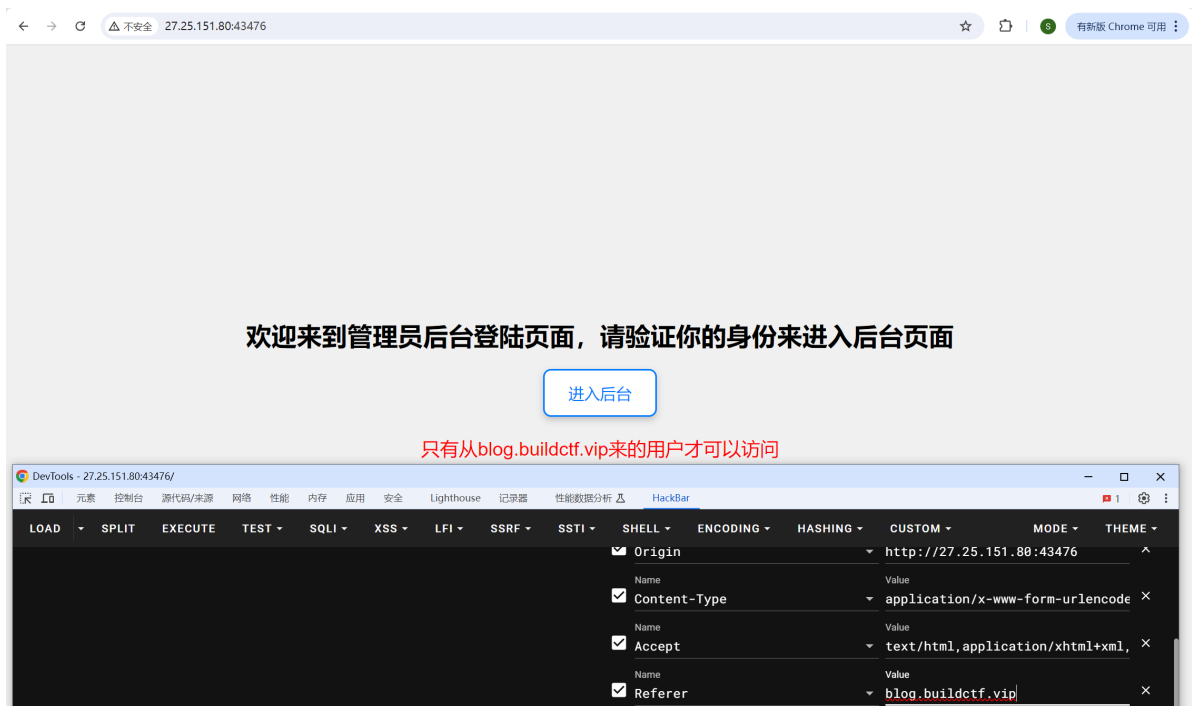


ez!http

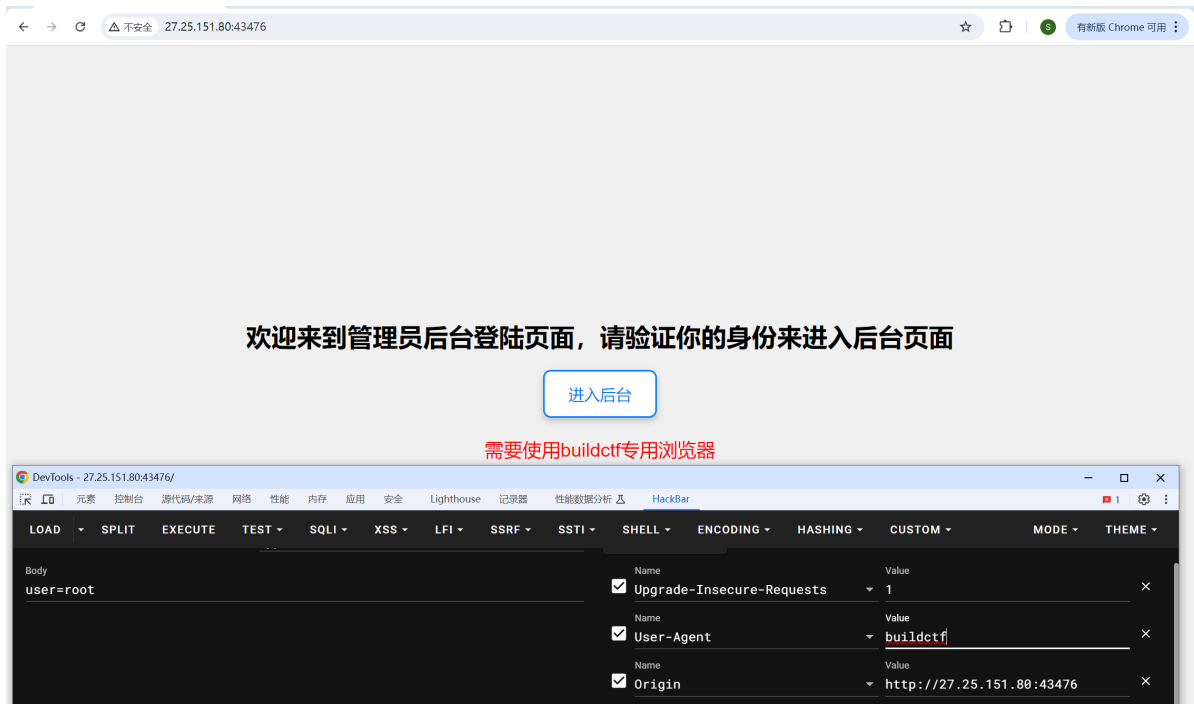
post传参user=root



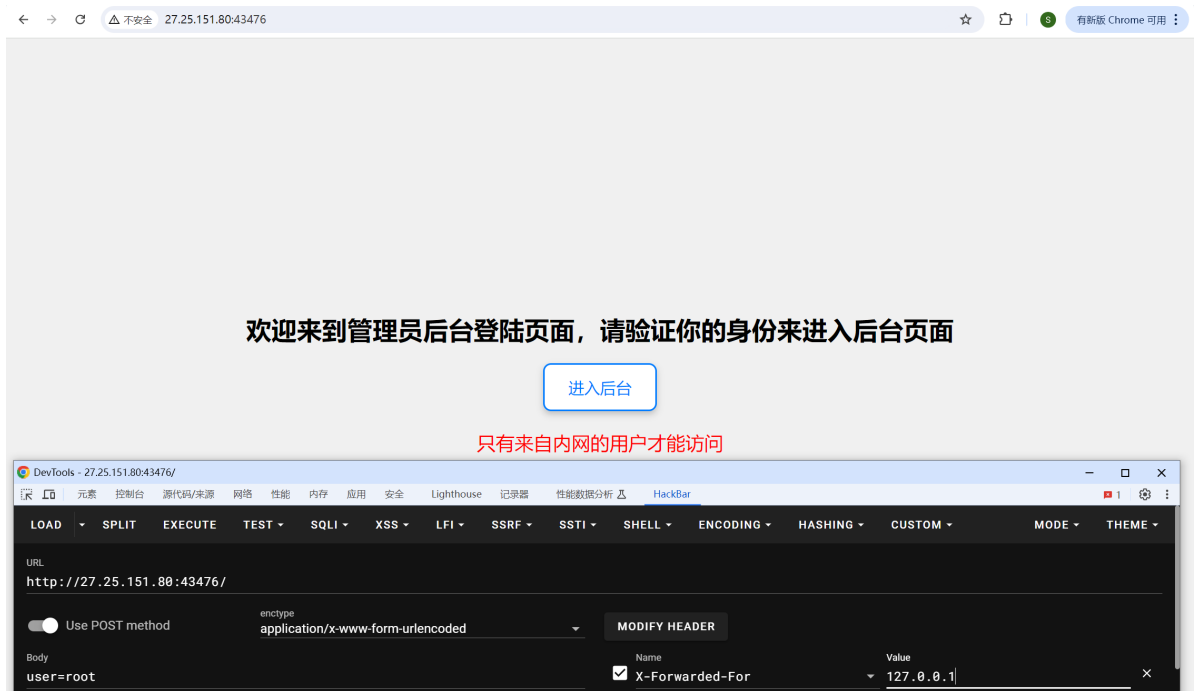
改referer头



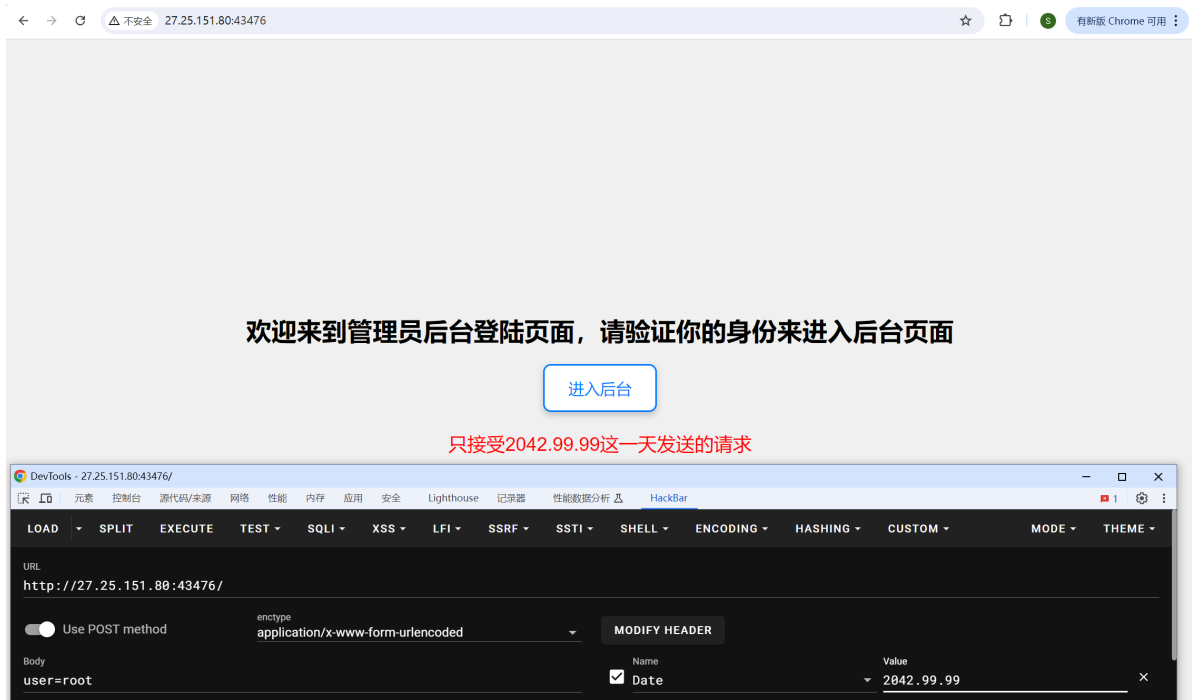
改UA头



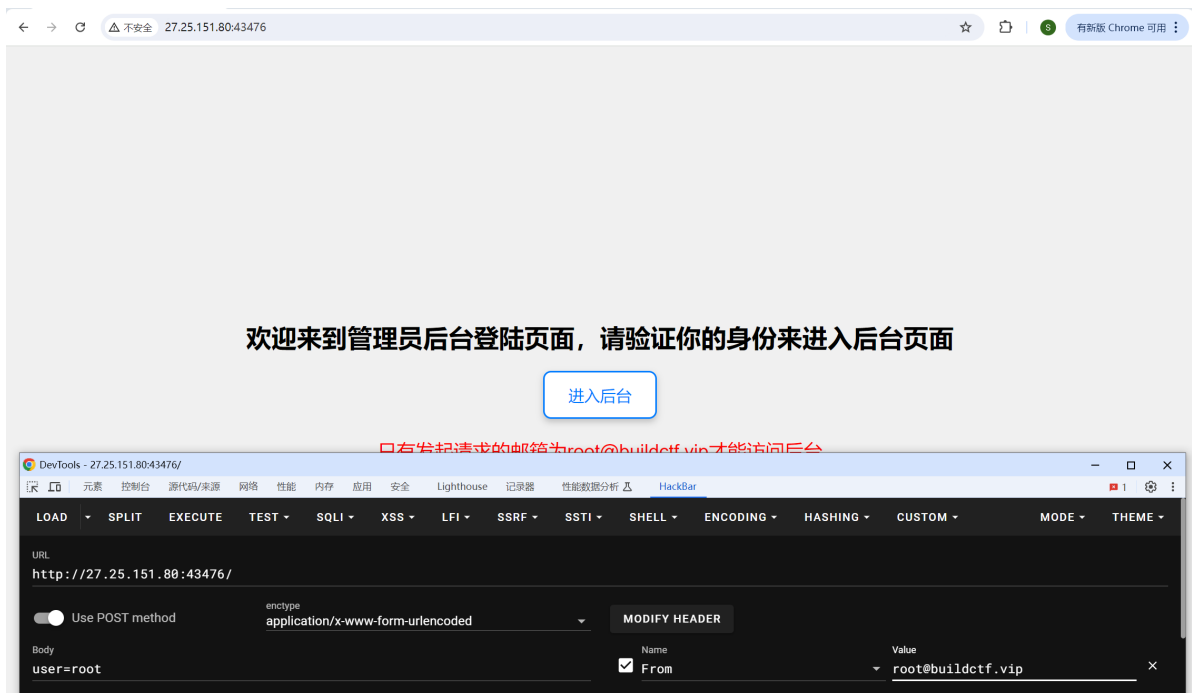
改XFF头



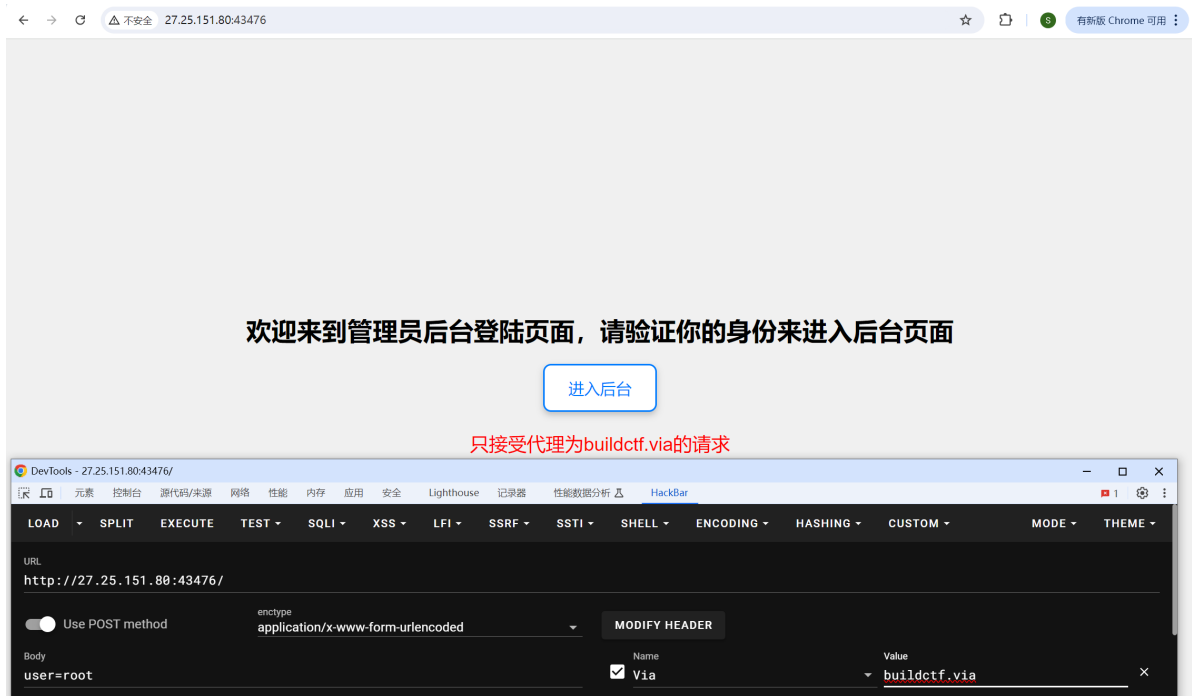
改Date头



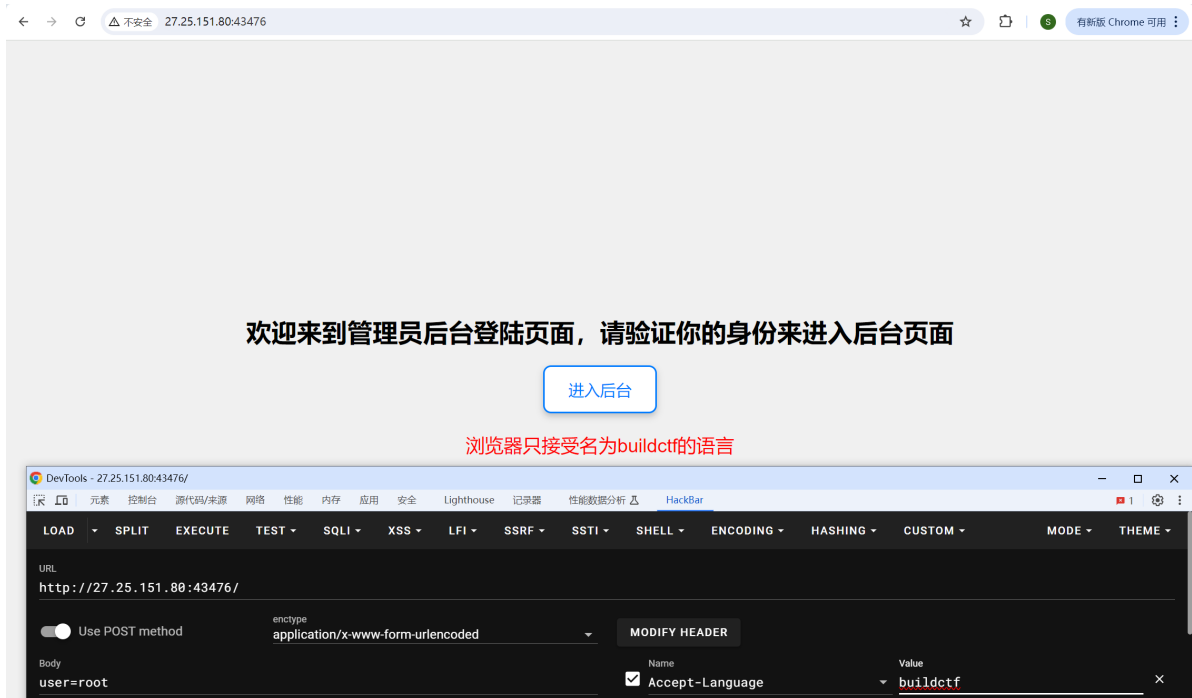
改from头



改via头



改AL头



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>
```

http://27.25.151.80:43476/

POST http://27.25.151.80:43476/ Send

Params Authorization Headers (15) Body Scripts Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

Key	Value	Description	Bulk Edit
☒ user	root		
☒ getFlag	This_is_flag		
Key	Value	Description	

Body Cookies Headers (6) Test Results 200 OK 86 ms 2.31 KB Save Response

Pretty Raw Preview Visualize

成功进入后台，欢迎您！
那我缺的flag在哪呢？点这个按钮试试
获取Flag
BuildCTF{e518785b-be7d-4054-8c79-fc2e4d149f12}

Postbot Runner Start Proxy Cookies Vault Trash

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

```
2 # visit https://tool.lu/pyc/ for more information
3 # Version: Python 3.8
4
5 import base64
6
7 def encode(message):
8     s = bytearray()
9     for i in message:
10         x = ord(i) ^ 32
11         x = x + 16
12         if x > 255:
13             x -= 256
14         s.append(x)
15     return base64.b64encode(bytes(s)).decode('utf-8')
16
17 correct = 'cmVZXFRzhHZrYFNpJyFj1VRVWmPV19ij4kgZW0='
18 flag = input(' Input flag: ')
19 if encode(flag) == correct:
20     print(' 正确的回答,awa!!!')
21 else:
22     print(' 就差一点了,QWQ!!')
23
```

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 = 'cmVZXFRzhHZrYFNpJyFj1VRVWmPV19ij4kgZW0='
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}
```