weChatDB

端口扫描

```python
└─# nmap -sV -A 192.168.56.104
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-12-02 09:50 CST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.104
Host is up (0.00068s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 10.0 (protocol 2.0)
80/tcp open  http    nginx
| http-robots.txt: 1 disallowed entry
|_/00000000000000000000000000000000/1.txt
|_http-title: QQ Group:660930334
MAC Address: 08:00:27:18:60:47 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.68 ms 192.168.56.104

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.21 seconds
```

有个 /00000000000000000000000000000000/1.txt，访问一下

```
aW9kant6aG9mcg==
```

base64 --> Caesar 得到 flag{welco

80 端口发现 rsa 数据，使用 yafu 分解 n 得到 p 和 q

```
YAFU Version 2.11
Built with Microsoft Visual Studio 1934 and Intel Compiler 2021
```

```
Using GMP-ECM 7.0.6-dev, Powered by MPIR 3.0.0
Detected 11th Gen Intel(R) Core(TM) i7-1180OH @ 2.30GHz
Detected L1 = 49152 bytes, L2 = 25165824 bytes, CL = 64 bytes
CPU features enabled: SSE41 AVX2 BMI2 AVX512F AVX512BW
Using 1 random witness for Rabin-Miller PRP checks
Cached 664579 primes; max prime is 9999991
Parsed yafu.ini from E:\CTFtoos\crypto\yafu

===============================================================
======= Welcome to YAFU (Yet Another Factoring Utility) =======
=======             bbuhrow@gmail.com                   =======
=======     Type help at any time, or quit to quit      =======
===============================================================

>>
factor(695213614751622239008466798517335821785048478231076660010312963576116553599228196690026107584614084172919726147096704484533476701096797379931545104638371750128092345556813521903264286265220947838315823011521222356097979992618819141838336099177371135667224478140001637751914435804937536947373393178888222136537411020638928353935326557965734754182508854607222474978989073706185059310115821938579075740438573327764798094594511834969283878350682304626852095672686682103548146686703763756983212214197396391922067377693886772083783856214431374008038233582058874902840530292468490451995314198441087608987360278558914219211257619500276728984703713564112049467783487247289962792652891730714007107964172466101623828565291735353365356169280458894278536144137217035254693720786235639605966882215820583744411548031939184665337480447385935203708179460117448169972048819301893758636728849545902355264490379152245758567142983052084604270386209677598316693019140278618909207620633805622023051743544497062638211794001178800133483562462265128833308421114792218003604856070025377192262196495961640424073025144214578561901418785199618425022926165437499416225038758030768086658881702811141409412059569388946479095015348160392431095827465549218283016325947885930054920796619628522569737310660068742309998028596616747069328197491100569396631603747310591754124509927698389278223136137719640573099037250022596830270350926924631251275932286130083365423314444082005270653273437735829165998746052007129587561783366231024800339220988424943846976282718515598724292452640995643707146797336985404425246597830582584058231956721891796339378982924557674592095115095120712487868706468508835988785175842229770708513382325637930110631453530080935110653985264984642541323345909371882248561225039828891633339047084511990403051046176880925865995100158541394330480603446263170138080277991274553248870278870283305114801351707934822276722978006821350876624647338541625182652350978801341649707490817500400625113245089148200255775950894547935942756903081360083615803912784310232125831086512581262812526429348426538118316522432607101029177879072419142166410516535564812733976209235944666954575218117663215948611208389459486792484281479485348676950126217558716370257319575449307597199196501876150634837505727105974474466182773069381558245644660037981122927112056577072496129748416207800697426228158426479600730742802892976980858615960773550681542591602087869824955779097840187912367791356566048863668450361448746 51)

fac: factoring
```

```
695213614751622239008466798517335821785048478231076660010312963576116553599228196690026107584614084172919726147096704484533476701096797379931545104638371750128092345556813521903264286265220947838315823011521222356097979992618819141838336099177371135667224478140001637751914435804937536947373393178888222136537411020638928353935326557965734754182508854607222474978989073706185059310115821938579075740438573327764798094594511834969283878350682304626852095672686682103548146686703763756983212214197396391922067377693886772083783856214431374008038233582058874902840530292468490451995314198441087608987360278558914219211257619500276728984703713564112049467783487247289962792652891730714007107964172466101623828565291735353365356169280458894278536144137217035254693720786235633960596688221582058374441154803193918466533748044738593520370817946011744816997204881930189375863672884954590235526449037915224575856714298305208460427038620967759831669301914027861890920762063380562202305174354449706263821179400117880013348356246226512883330842111479221800360485607002537719226219649596164042407302514421457856190141878519961842502292616543749941622503875803076808665888170281114140941205956938894647909501534816039243109582746554921828301632594788593005492079661962852256973731066006874230999802859661674706932819749110056939663160374731059175412450992769838927822313613771964057309903725002259683027035092692463125127593228613008336542331444408200527065327343773582916599874605200712958756178336623102480033922098842494384697628271851559872429245264099564370714679733698540442524659783058258405823195672189179633937898292455767459209511509512071248786870646850883598878517584222977070851338232563793011063145353008093511065398526498464254132334590937188224856122503982889163333904708451199040305104617688092586599510015854139433048060344626317013808027799127455324887027887028330511480135170793482227672297800682135087662464733854162518265235097880134164970749081750040062511324508914820025577595089454793594275690308136008361580391278431023212583108651258126281252642934842653811831652243260710102917787907241914216641051653556481273397620923594466695457521811766321594861120838945948679248428147948534867695012621755871637025731957544930759719919650187615063483750572710597447446618277306938155824564466003798112292711205657707249612974841620780069742622815842647960073074280289297698085861596077355068154259160208786982495577909784018791236779135656604886366845036144874651
fac: using pretesting plan: normal
fac: no tune info: using qs/gnfs crossover of 100 digits
fac: no tune info: using qs/snfs crossover of 75 digits
div: primes less than 10000
fmt: 1000000 iterations
Total factoring time = 0.7414 seconds


***factors found***

PRP1233 =
833794707797802131749427474533851285168506931260624622649677278729595097385552136066205433738568617376547766764117013537419705792663474151700197357522609690110959943683328167871932720812937790702139291479651858573233204862863903968709873364261031952013602765604234735018337192056707396676174267749917115413253324600741895467609009046822407281907886070139271779218107400470953261223928057471535253796316622036133618316747190532483828649854206155137155924060353280889063353194361446
```

```
13838118496563556419984503457791746028254931047517256521726259836993949926887562
67774135139384269326790217145422467438090296833426700968334326195144022677636108
90150232993199917215696048666192006834705668266063870593891468063941484318957583
10088527486881359800728834869272890323370403438880053907524027223497239609777922
87423273853953801828331190401091782581749147813186641110361321847609799710020213
69704524809203075230974226033129315396033767474152293898206533544589281356248772
14178698552264887324255066935009684675218868578340873786262384484664079715148099
77172011173553703271095360313341934455487999887263463188158818745762681642685779
94050902444925043233820668857043905798324268404244448510404940155169472351169094
09005269050521477819424571854030 1
PRP1233 =
83379470779780213174942747453385128516850693126062462264967727872959509738555213
60662054337385686173765477667641170135374197057926634741517001973575226096901109
59943683328167871932720812937790702139291479651858573233204862863903968709873364
26103195201360276566042347350183371920567073966761742677499171154132533246007418
95467609009046822407281907886070139271779218107400470953261223928057471535253796
31662203613361831674719053248382864985420615513715592406035328088906335319436144
13838118496563556419984503457791746028254931047517256521726259836993949926887562
67774135139384269326790217145422467438090296833426700968334326195144022677636108
90150232993199917215696048666192006834705668266063870593891468063941484318957583
10088527486881359800728834869272890323370403438880053907524027223497239609777922
87423273853953801828331190401091782581749147813186641110361321847609799710020213
69704524809203075230974226033129315396033767474152293898206533544589281356248772
14178698552264887324255066935009684675218868578340873786262384484664079715148099
77172011173553703271095360313341934455487999887263463188158818745762681642685779
94050902444925043233820668857043905798324268404244448510404940155169472351169094
09005269050521477819424571852935 1

ans = 1

>>
```

解一个 rsa

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

c =
24857232969597347836205218238683593974178064734529537038494876506626461580229609
00498151880364315766317977474341051065573476331176500505213212100453368995447255
07571957071600903886005417706294793737099741836900763694331987804822291585926732
61637783024448643762634374608937542828739538789886353060899749093911968298945799
54774591949590479126005312033500431806844243471799888730579915067616252753501953
44920627652206158301633645063851914041020965562368988326399931096094011388686864
69245937198539144609594020948220868710728246463334051834284576670141046263742583
43868912400087509780210950016417836287906555772474208992044934956618014472015700
```

989714190014383201041575706971847243818333408671036047210269957218785025095037478813679891591208104332861012889816048117826153501411993345193043557168913320235820007854491320190564266699613759700930956831687443279337761217343101824804966745536330395323663256870608145672789228321991171708880967049812107848524541445751307274087560169716194303372219345735317742584989145143531366576099302175008964181211685968251896719222208354871385483687055255400760571529007806202779961984080503046589672012884211371902876284574015690726077969666760372434140377004249202141982748179132634988493779524872578782792131778392374262306449798895881334657755837931706831809672301189285295998120682686889889828119209105485185277100624833127751042037721088305714175995236977010639473655562183826537495742343998816739551903633602103236130355168651905238223946918646061588167045864707020822037544002096869654702374297651421584999266164991033816603084782178256718232022558737987349351991007150060455439616659753673593909416371810346543457149267615552148190258952123107729814993315951988727889414099267695042829143002072191860246710883073749224700926364999175308264029285369879486929420038562931750940469488374702685225942836047952246713360146328068454725731735728058310179365632521887814232719311030933932891276869584247773371232397986996460845366123942297634622317849081098316733155104237504021685543680537749039667632121196027261239470320160618535372879133317486970850833133645366663198139687269252488349072591702588561551062784869631175082890879270725588361411296180872685859968589983256039368218538660061609169373882567676270935167544464204536997509428839711265345764661097800267922487615358939431948789940088886538762300530013662837482353178316426185404480339102388989936553514117468058944041640372958139818451214604535417344832379778506856166489745
n =
695213614751622239008466798517335821785048478231076660010312963576116553599228196690026107584614084172919726147096704484533476701096797379931545104638371750128092345556813521903264286265220947838315823011521222356097979992618819141838336099177371135667224478140001637751914435804937536947373393178888222136537411020638928353935326557965734754182508854607222474978989073706185059310115821938579075740438573327764798094594511834969283878350682304626852095672686682103548146686703763756983212214197396391922067377693886772083783856214431374008038233582058874902840530292468490451995314198441087608987360278558914219211257619500276728984703713564112049467783487247289962792652891730714007107964172466101623828565291735353365356169280458894278536144137217035254693720786235633960596688221582058374441154803193918466533748044738593520370817946011744816997204881930189375863672884954590235526449037915224575856714298305208460427038620967759831669301914027861890920762063380562202305174354449706263821179400117880013348356246226512883330842111479221800360485607002537719226219649596164042407302514421457856190141878519961842502292616543749941622503875803076808665888170281114140941205956938894647909501534816039243109582746554921828301632594788593005492079661962852256973731066006874230999802859661674706932819749110056939663160374731059175412450992769838927822313613771964057309903725002259683027035092692463125127593228613008336542331444408200527065327343773582916599874605200712958756178336623102480033922098842494384697628271851559872429245264099564370714679733698540442524659783058258405823195672189179633937898292455767459209511509512071248786870646850883598878517584222977070851338232563793011063145353008093511065398526498464254132334590937188224856122503982889163333904708451199040305104617688092586599510015854139433048060344626317013808027799127455324887027887028330511480135170793482227672297800682135087662464733854

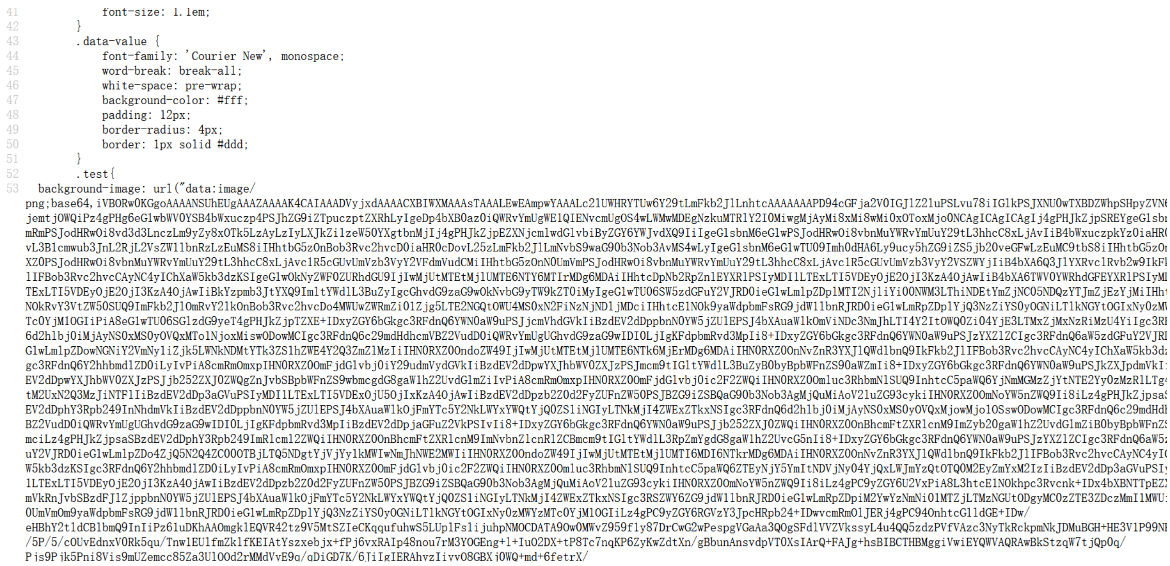
```
162518265235097880134164970749081750040062511324508914820025577595089454793594275690308136008361580391278431023212583108651258126281252642934842653811831652243260710102917787907241914216641051653556481273397620923594466695457521811766321594861120838945948679248428147948534867695012621755871637025731957544930759719919650187615063483750572710597447446618277306938155824564466003798112292711205657707249612974841620780069742622815842647960073074280289297698085861596077355068154259160208786982495577909784018791236779135656604886366845036144874651
e = 65537


p =
833794707797802131749427474533851285168506931260624622649677278729595097385552136066205433738568617376547766764117013537419705792663474151700197357522609690110959943683328167871932720812937790702139291479651858573233204862863903968709873364261031952013602765660423473501833719205670739667617426774991711541325332460074189546760900904682240728190788607013927177921810740047095326122392805747153525379631662203613361831674719053248382864985420615513715592406035328088906335319436144138381184965635564199845034577917460282549310475172565217262598369939499268875626777413513938426932679021714542246743809029683342670096833432619514402267763610890150232993199917215696048666192006834705668266063870593891468063941484318957583100885274868813598007288348692728903233704034388800539075240272234972396097779228742327385395380182833119040109178258174914781318664111036132184760979971002021369704524809203075230974226033129315396033767474152293898206533544589281356248772141786985522648873242550669350096846752188685783408737862623844846640797151480997717201117355370327109536031334193445548799988726346318815881874576268164268577994050902444925043233820668857043905798324268404244448510404940155169472351169094090052690505214778194245718540301
q =
833794707797802131749427474533851285168506931260624622649677278729595097385552136066205433738568617376547766764117013537419705792663474151700197357522609690110959943683328167871932720812937790702139291479651858573233204862863903968709873364261031952013602765660423473501833719205670739667617426774991711541325332460074189546760900904682240728190788607013927177921810740047095326122392805747153525379631662203613361831674719053248382864985420615513715592406035328088906335319436144138381184965635564199845034577917460282549310475172565217262598369939499268875626777413513938426932679021714542246743809029683342670096833432619514402267763610890150232993199917215696048666192006834705668266063870593891468063941484318957583100885274868813598007288348692728903233704034388800539075240272234972396097779228742327385395380182833119040109178258174914781318664111036132184760979971002021369704524809203075230974226033129315396033767474152293898206533544589281356248772141786985522648873242550669350096846752188685783408737862623844846640797151480997717201117355370327109536031334193445548799988726346318815881874576268164268577994050902444925043233820668857043905798324268404244448510404940155169472351169094090052690505214778194245718529351

print(p*q==n)

d = gmpy2.invert(e, (p-1)*(q-1))
```

```
m = pow(c,d,n)
print(long_to_bytes(m))
```

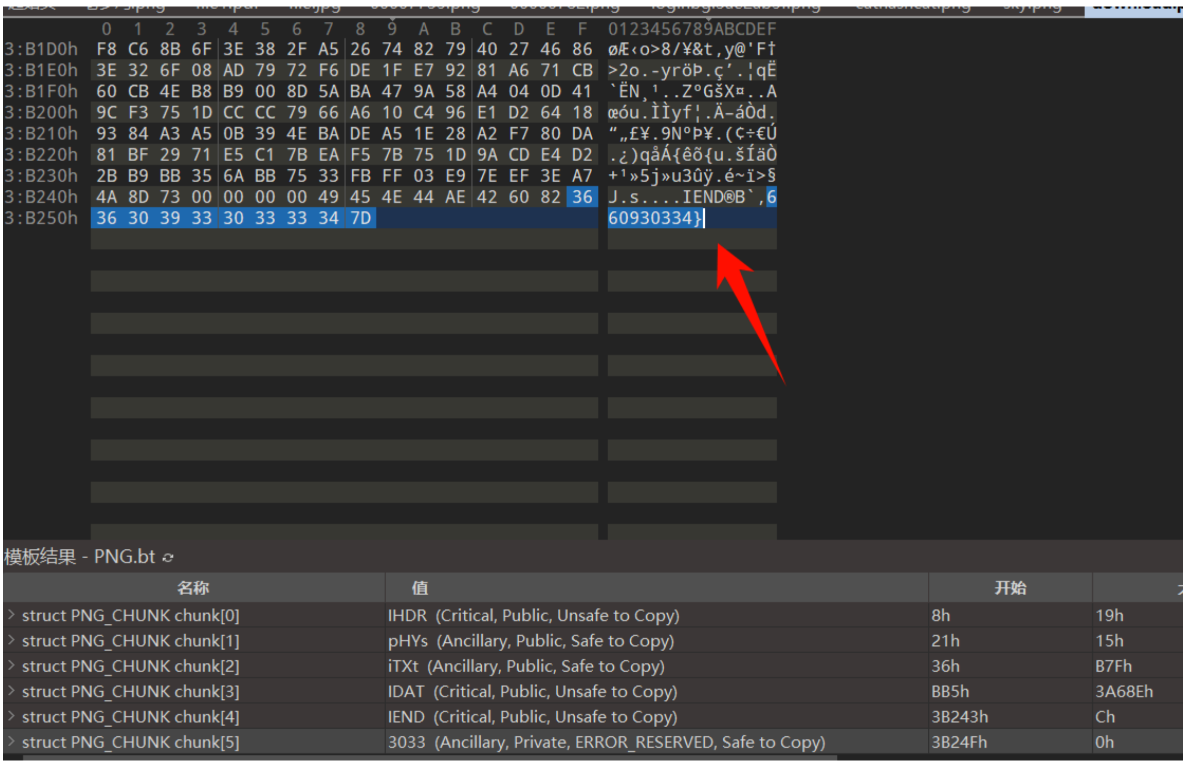
得到

```
me:wlc0mE@
```

查看页面源代码，发现有一张图片，但是页面并没有显示出来，提取一下



发现后半段 flag



660930334}

组和一下得到，flag{welcome:wlc0mE@660930334} 尝试 ssh 登入

```
ssh welcome@192.168.56.104
```

flag：flag{user-415621D5297F8F4BE138A5BB03}

先打包一下 wechat_files

```
# 打包 wechat_files
tar -czf wechat_files.tar.gz wechat_files
```

然后把文件搞下来

```
scp welcome@192.168.56.104:~/wechat_files.tar.gz .
```

在 wechat_files\lingdong\msg 下面发现 MSG0.db 应该是要解密一下，找到一篇文章微信PC端数据库文件解密_微信数据库解密-CSDN博客

写个解密脚本解密 MSG0.db

```
#!/usr/bin/env python3
import ctypes
import hashlib
```

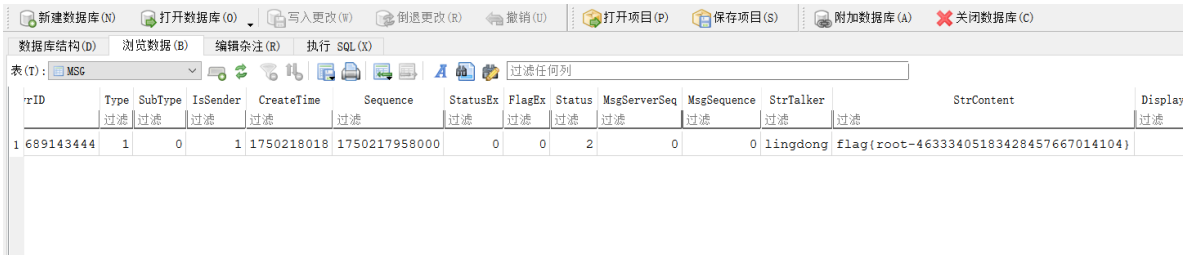
```python
import hmac
from pathlib import Path
from Crypto.Cipher import AES

# 常量
SQLITE_FILE_HEADER = bytes('SQLite format 3', encoding='ASCII') + bytes(1)
KEY_SIZE = 32
DEFAULT_PAGESIZE = 4096
DEFAULT_ITER = 64000

# 配置
DB_FILE = r"wechat_files\lingdong\msg\MSG0.db"
KEY_HEX = "c22ce55044354439b22d75a1e1e4be286bc480cde0f34583bb490fe686b56061"
OUTPUT_DB = r"wechat_files\lingdong\msg\decoded_MSG0.db"

def decrypt_database():
    # 转换密钥
    password = bytes.fromhex(KEY_HEX)

    # 读取数据库
    with open(DB_FILE, 'rb') as f:
        blist = f.read()

    # 提取盐值（前16字节）
    salt = blist[:16]

    # 派生密钥
    key = hashlib.pbkdf2_hmac('sha1', password, salt, DEFAULT_ITER, KEY_SIZE)

    # 获取第一页
    first = blist[16:DEFAULT_PAGESIZE]

    # 验证 HMAC
    mac_salt = bytes([x ^ 58 for x in salt])
    mac_key = hashlib.pbkdf2_hmac('sha1', key, mac_salt, 2, KEY_SIZE)
    hash_mac = hmac.new(mac_key, digestmod='sha1')
    hash_mac.update(first[:-32])
    hash_mac.update(bytes(ctypes.c_int(1)))

    if hash_mac.digest() != first[-32:-12]:
        print("密钥错误！")
        return False

    print("HMAC 验证成功！")

    # 分页
    pages = [blist[i:i + DEFAULT_PAGESIZE]
             for i in range(DEFAULT_PAGESIZE, len(blist), DEFAULT_PAGESIZE)]
```

```
    # 解密并写入
    with open(OUTPUT_DB, 'wb') as f:
        # 写入 SQLite 文件头
        f.write(SQLITE_FILE_HEADER)

        # 解密第一页
        cipher = AES.new(key, AES.MODE_CBC, first[-48:-32])
        f.write(cipher.decrypt(first[:-48]))
        f.write(first[-48:])

        # 解密其他页
        for page in pages:
            if len(page) == DEFAULT_PAGESIZE:
                cipher = AES.new(key, AES.MODE_CBC, page[-48:-32])
                f.write(cipher.decrypt(page[:-48]))
                f.write(page[-48:])
            else:
                f.write(page)

    print(f"解密成功! 输出: {OUTPUT_DB}")
    return True

if __name__ == '__main__':
    decrypt_database()
```

然后读取一下数据库中的数据

```
import sqlite3

conn = sqlite3.connect('decoded_MSGO.db')
c = conn.cursor()

c.execute("SELECT name FROM sqlite_master WHERE type='table'")
tables = c.fetchall()
print("表:", tables)

c.execute("PRAGMA table_info(MSG)")
columns = c.fetchall()
print("MSG 表字段:", columns)

c.execute("SELECT * FROM MSG")
rows = c.fetchall()
for row in rows:
    print(row)
```

或者直接使用 DB Browser for SQLCipher 这个工具查看



flag：flag{root-463334051834284576670141O4}