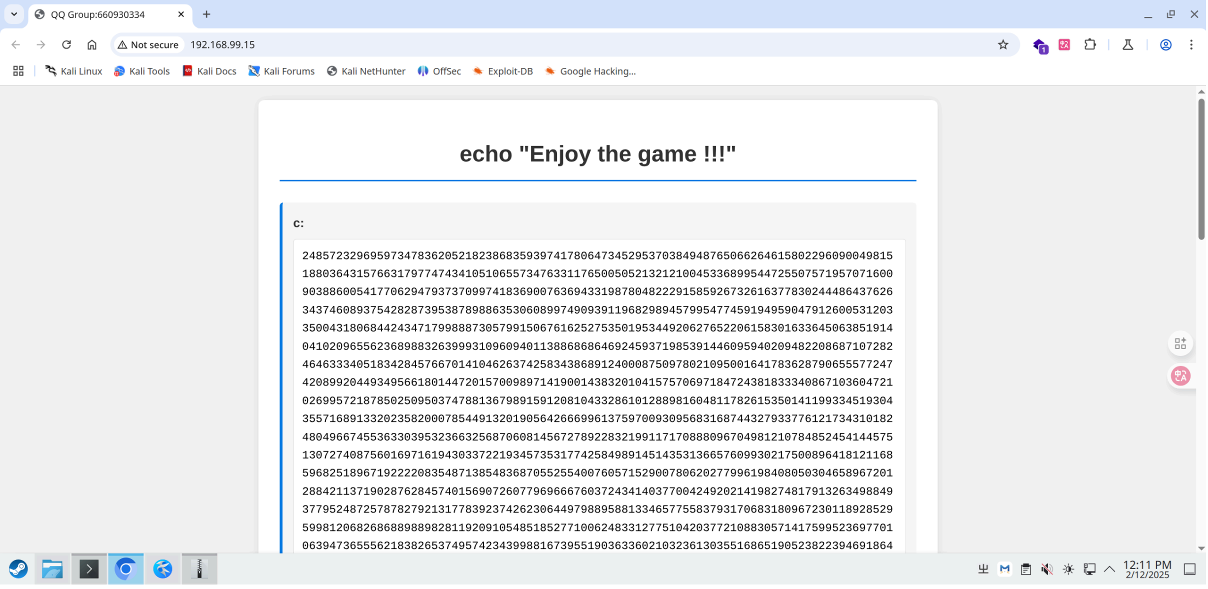
weChatDB

0. Scan

```
➜  WeChatDB rustscan -a 192.168.99.15
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.
________________________________________
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Please contribute more quotes to our GitHub https://github.com/rustscan/rustscan

[~] The config file is expected to be at "/home/user/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping with --ulimit. May cause harm to sensitive servers
[!] Your file limit is very small, which negatively impacts RustScan's speed. Use the Docker image, or up the Ulimit with '--ulimit 5000'.
Open 192.168.99.15:22
Open 192.168.99.15:53
Open 192.168.99.15:80
[~] Starting Script(s)
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-02 12:10 HKT
Initiating ARP Ping Scan at 12:10
Scanning 192.168.99.15 [1 port]
Completed ARP Ping Scan at 12:10, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:10
Completed Parallel DNS resolution of 1 host. at 12:10, 0.00s elapsed
DNS resolution of 1 IPs took 0.00s. Mode: Async [#: 3, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
```

```
Initiating SYN Stealth Scan at 12:10
Scanning 192.168.99.15 [3 ports]
Discovered open port 22/tcp on 192.168.99.15
Discovered open port 80/tcp on 192.168.99.15
Completed SYN Stealth Scan at 12:10, 0.02s elapsed (3 total ports)
Nmap scan report for 192.168.99.15
Host is up, received arp-response (0.00031s latency).
Scanned at 2025-12-02 12:10:40 HKT for 0s

PORT   STATE  SERVICE REASON
22/tcp open   ssh     syn-ack ttl 64
53/tcp closed domain  reset ttl 64
80/tcp open   http    syn-ack ttl 64
MAC Address: 08:00:27:79:85:F5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
           Raw packets sent: 4 (160B) | Rcvd: 4 (156B)
```

只开了 22 和 80

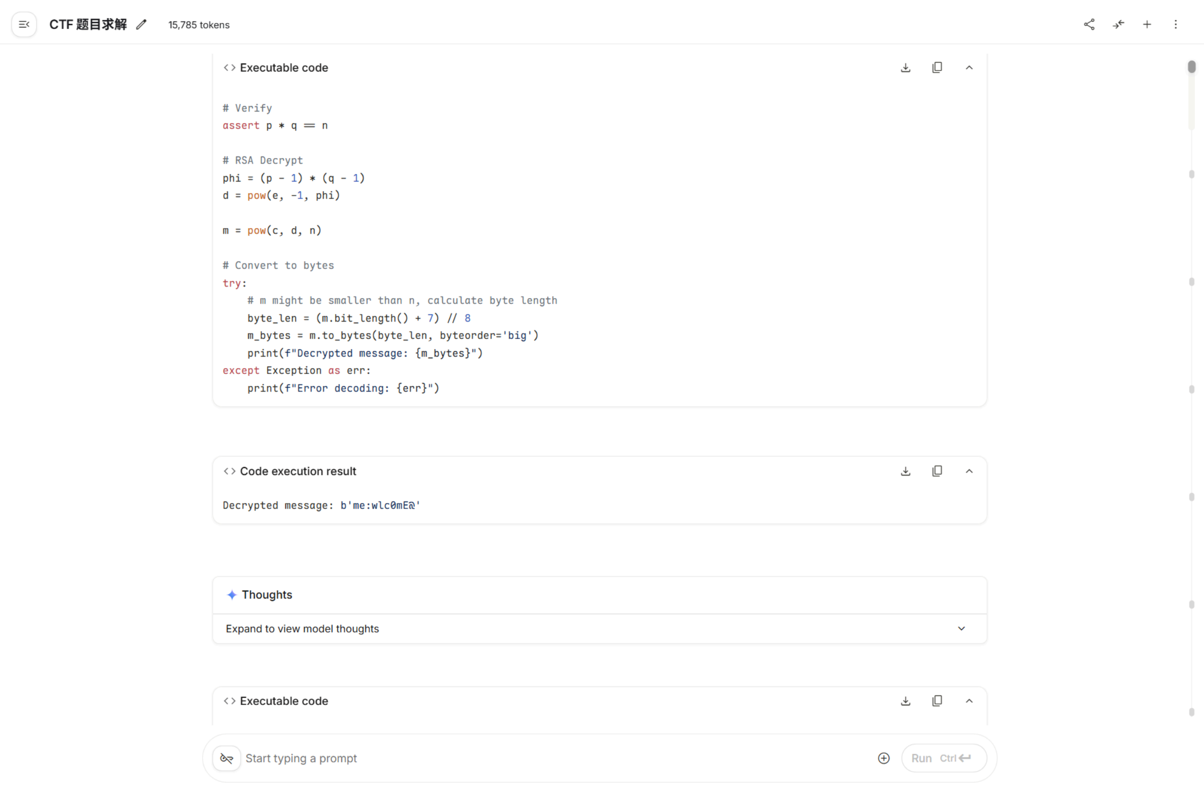


echo "Enjoy the game !!!"

c:

```
248572329695973478362052182386835939741780647345295370384948765066264615802296090049815
```

Web 有 c、n、e，RSA。在线查询没查到，Gemini 3 Pro 秒了



```
# Verify
assert p * q == n

# RSA Decrypt
phi = (p - 1) * (q - 1)
d = pow(e, -1, phi)

m = pow(c, d, n)

# Convert to bytes
try:
    # m might be smaller than n, calculate byte length
    byte_len = (m.bit_length() + 7) // 8
    m_bytes = m.to_bytes(byte_len, byteorder='big')
    print(f"Decrypted message: {m_bytes}")
except Exception as err:
    print(f"Error decoding: {err}")
```

```
Decrypted message: b'me:wlc0mE@'
```

```
me:wlc0mE@
```

看起来像用户名和密码，但是无法登录

继续扫描目录

```
➜ WeChatDB ffuf -w /usr/share/wordlists/SecLists/Discovery/Web-Content/big.txt -e .php,.html,.txt -u http://192.168.99.15/FUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://192.168.99.15/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/SecLists/Discovery/Web-Content/big.txt
 :: Extensions       : .php .html .txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

index.html              [Status: 200, Size: 330191, Words: 761, Lines: 84, Duration: 1ms]
robots.txt              [Status: 200, Size: 63, Words: 3, Lines: 2, Duration: 1ms]
robots.txt              [Status: 200, Size: 63, Words: 3, Lines: 2, Duration: 1ms]
:: Progress: [81912/81912] :: Job [1/1] :: 20000 req/sec :: Duration: [0:00:03] :: Errors: 0 ::
```

发现 robots.txt

```
➜  WeChatDB curl -v http://192.168.99.15/robots.txt
*   Trying 192.168.99.15:80...
* Connected to 192.168.99.15 (192.168.99.15) port 80
* using HTTP/1.x
> GET /robots.txt HTTP/1.1
> Host: 192.168.99.15
> User-Agent: curl/8.14.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Server: nginx
< Date: Tue, 02 Dec 2025 04:14:18 GMT
< Content-Type: text/plain
< Content-Length: 63
< Last-Modified: Sat, 29 Nov 2025 06:23:08 GMT
< Connection: keep-alive
< ETag: "692a914c-3f"
< Accept-Ranges: bytes
```

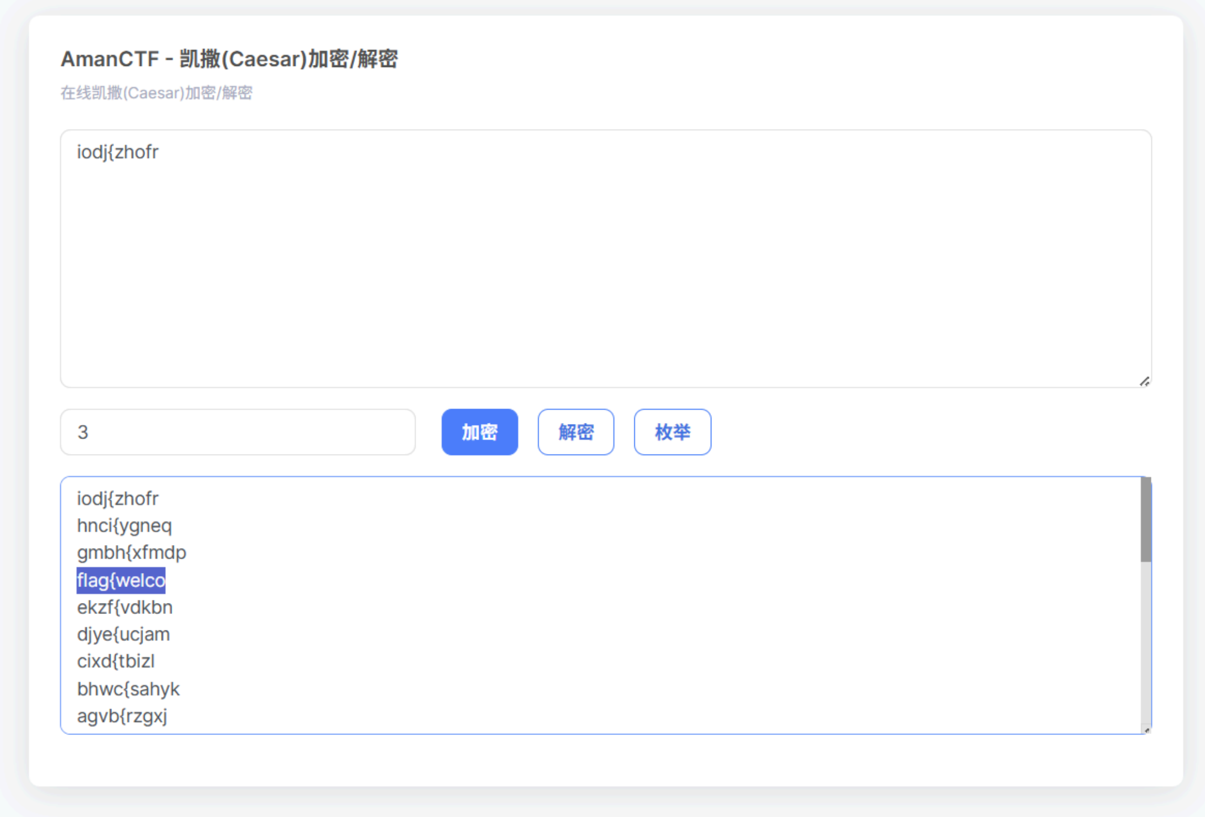
```
<
User-agent: *
* Connection #0 to host 192.168.99.15 left intact
Disallow: /00000000000000000000000000000000/1.txt
```

```
➜  WeChatDB curl -v http://192.168.99.15/00000000000000000000000000
000000/1.txt
*   Trying 192.168.99.15:80...
* Connected to 192.168.99.15 (192.168.99.15) port 80
* using HTTP/1.x
> GET /00000000000000000000000000000000/1.txt HTTP/1.1
> Host: 192.168.99.15
> User-Agent: curl/8.14.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Server: nginx
< Date: Tue, 02 Dec 2025 04:14:50 GMT
< Content-Type: text/plain
< Content-Length: 16
< Last-Modified: Sat, 29 Nov 2025 06:22:59 GMT
< Connection: keep-alive
< ETag: "692a9143-10"
< Accept-Ranges: bytes
<
* Connection #0 to host 192.168.99.15 left intact
aW9kant6aG9mcg==
```

发现 Base64

```
➜  WeChatDB echo "aW9kant6aG9mcg==" | base64 -d
iodj{zhofr
```

像凯撒



iodj{zhofr

3

```
iodj{zhofr
hnci{ygneq
gmbh{xfmdp
flag{welco
ekzf{vdkbn
djye{ucjam
cixd{tbizl
bhwc{sahyk
agvb{rzgxj
```

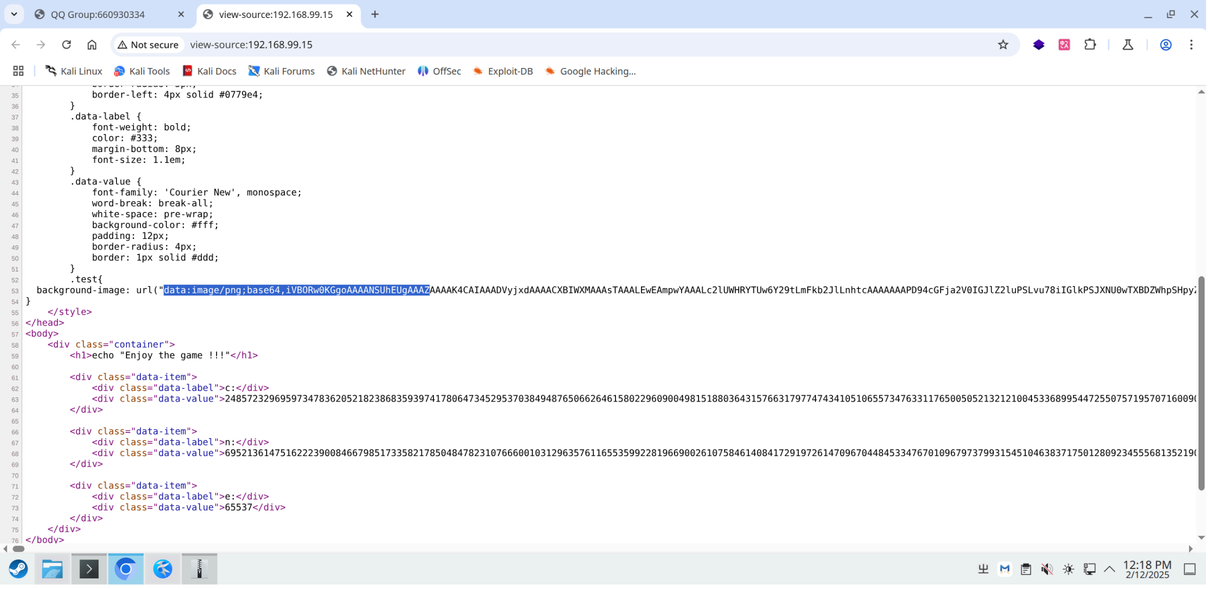
凯撒 3 层成功解密

现在就得到了

```
flag{welcome:wlc0mE@
```

使用 welcome 和密码登录，依旧不行。flag 还没有闭合，猜测还有别的东西没找到

之前复制 HTML 给 Gemini 的时候，有很长一串 Background，但是网页并没有用到 Background，猜测是有用的图片



```
            border-left: 4px solid #0779e4;
        }
        .data-label {
            font-weight: bold;
            color: #333;
            margin-bottom: 8px;
            font-size: 1.1em;
        }
        .data-value {
            font-family: 'Courier New', monospace;
            word-break: break-all;
            white-space: pre-wrap;
            background-color: #fff;
            padding: 12px;
            border-radius: 4px;
            border: 1px solid #ddd;
        }
        .test{
  background-image: url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAZAAAAK4CAIAAADVyjxdAAAACXBIWXMAAAsTAAALEwEAmpwYAAALc2lUWHRYTUw6Y29tLmFkb2JlLnhtcAAAAAAAPD94cGFja2V0IGJlZ2luPSLvu78iIGlkPSJXNU0wTXBDZWhpSHpy
}
    </style>
</head>
<body>
    <div class="container">
        <h1>echo "Enjoy the game !!!"</h1>

        <div class="data-item">
            <div class="data-label">c:</div>
            <div class="data-value">24857232969597347836205218238683593974178064734529537038494876506626461580229609004981518803643157663179774743410510655734763311765005052132121004533689954472550757195707160090
        </div>

        <div class="data-item">
            <div class="data-label">n:</div>
            <div class="data-value">69521361475162223900846679851733582178504847823107666001031296357611655359922819669002610758461408417291972614709670448453347670109679737993154510463837175012809234555681352190
        </div>

        <div class="data-item">
            <div class="data-label">e:</div>
            <div class="data-value">65537</div>
        </div>
    </div>
</body>
```

看了下隐写，没东西。试试 strings

```
azwhikaru@Hinana MINGW64 ~/Downloads
$ strings "./download (1).png"
IHDR
...
IEND
660930334}
```

发现闭合的括号

```
flag{welcome:wlc0mE@660930334}
```

1. Get Shell

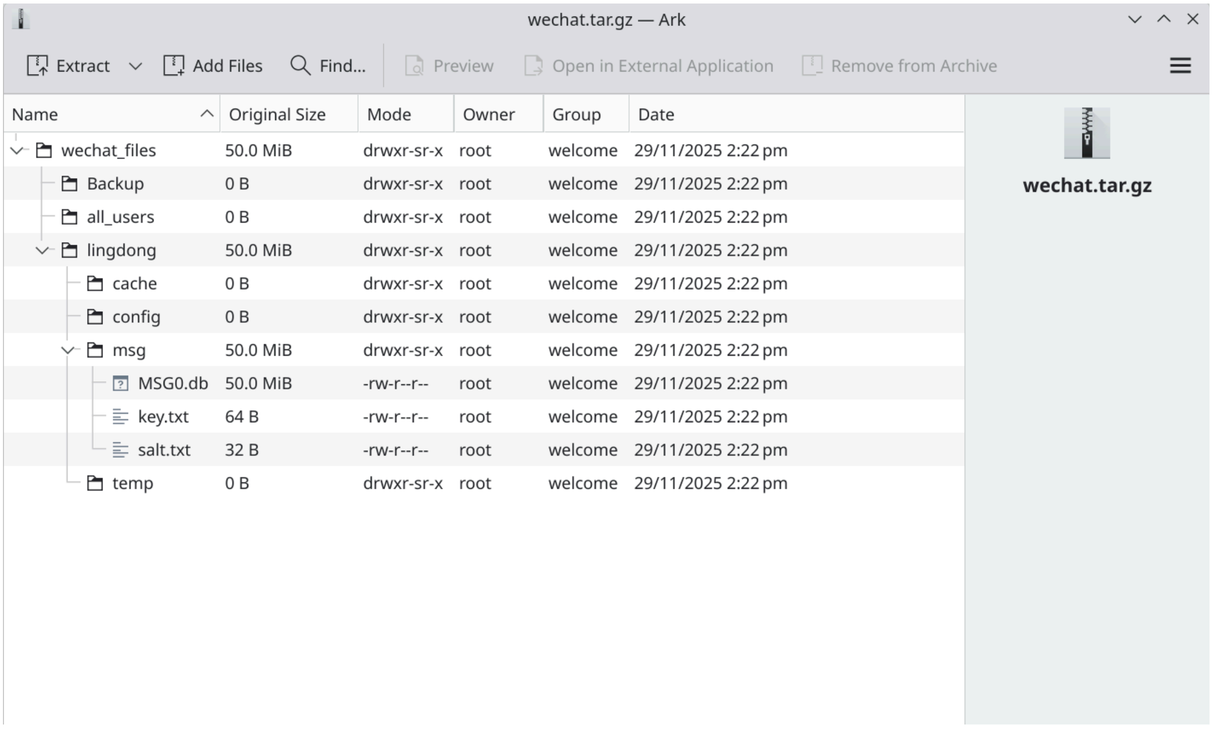
使用凭据登录

```
➜ WeChatDB ssh welcome@192.168.99.15
welcome@192.168.99.15's password:
================================
Welcome!!!
QQ Group:660930334
================================
lingdong:~$
```

```
lingdong:~$ ls -al
total 20
drwxr-sr-x    3 welcome  welcome       4096 Nov 29 14:22 .
drwxr-xr-x    3 root     root          4096 Jun  3 08:22 ..
lrwxrwxrwx    1 root     welcome          9 Jun  3 09:07 .ash_history → /dev/null
-rw-r--r--    1 root     welcome          6 Nov 29 14:22 tip.txt
-rw-r--r--    1 root     welcome         37 Nov 29 14:22 user.txt
drwxr-sr-x    5 root     welcome       4096 Nov 29 14:22 wechat_files
lingdong:~$ cat tip.txt
wechatlingdong:~$
```

提示 wechat。发现 wechat_files 文件夹，猜测是微信聊天记录取证

打包 tar 再用 nc 传出来



Name	Original Size	Mode	Owner	Group	Date
wechat_files	50.0 MiB	drwxr-sr-x	root	welcome	29/11/2025 2:22 pm
Backup	0 B	drwxr-sr-x	root	welcome	29/11/2025 2:22 pm
all_users	0 B	drwxr-sr-x	root	welcome	29/11/2025 2:22 pm
lingdong	50.0 MiB	drwxr-sr-x	root	welcome	29/11/2025 2:22 pm
cache	0 B	drwxr-sr-x	root	welcome	29/11/2025 2:22 pm
config	0 B	drwxr-sr-x	root	welcome	29/11/2025 2:22 pm
msg	50.0 MiB	drwxr-sr-x	root	welcome	29/11/2025 2:22 pm
MSG0.db	50.0 MiB	-rw-r--r--	root	welcome	29/11/2025 2:22 pm
key.txt	64 B	-rw-r--r--	root	welcome	29/11/2025 2:22 pm
salt.txt	32 B	-rw-r--r--	root	welcome	29/11/2025 2:22 pm
temp	0 B	drwxr-sr-x	root	welcome	29/11/2025 2:22 pm

发现微信数据库和 Key

按照网上方法解密 RawKey 再用 DB Browser 打开，只能看到表结构，看不到数据

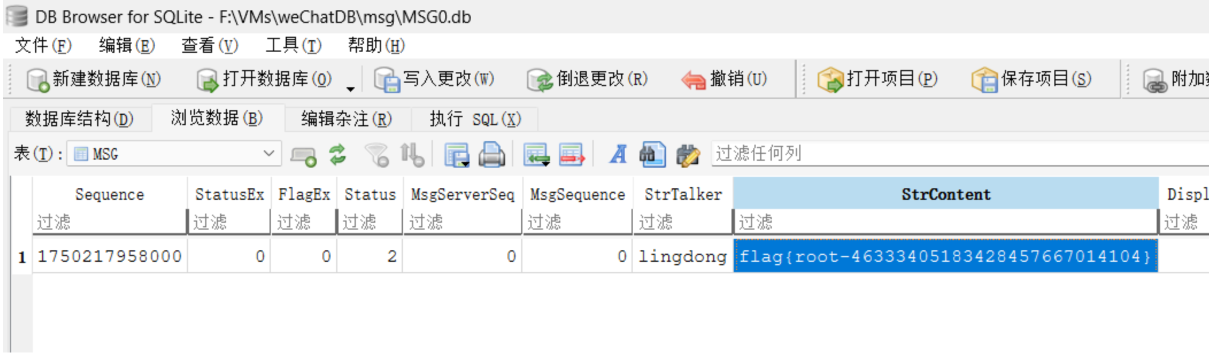
后来换了几个脚本，这个可以用

```
from Crypto.Cipher import AES
import hashlib, hmac, ctypes, sys
```

```
SQLITE_FILE_HEADER = bytes('SQLite format 3', encoding='ASCII') + bytes(1)
IV_SIZE = 16
HMAC_SHA1_SIZE = 20
KEY_SIZE = 32
DEFAULT_PAGESIZE = 4096
DEFAULT_ITER = 64000
input_pass = input('key:')
input_dir = input('file:')

password = bytes.fromhex(input_pass.replace(' ', ''))

with open(input_dir, 'rb') as (f):
    blist = f.read()
print(len(blist))
salt = blist[:16]
key = hashlib.pbkdf2_hmac('sha1', password, salt, DEFAULT_ITER, KEY_SIZE)
first = blist[16:DEFAULT_PAGESIZE]
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('Decryption Success')
else:
    print('Password Error')
blist = [blist[i:i + DEFAULT_PAGESIZE] for i in range(DEFAULT_PAGESIZE, len(blist), DEFAULT_PAGESIZE)]

with open(input_dir, 'wb') as (f):
    f.write(SQLITE_FILE_HEADER)
    t = AES.new(key, AES.MODE_CBC, first[-48:-32])
    f.write(t.decrypt(first[:-48]))
    f.write(first[-48:])
```

```
for i in blist:
    t = AES.new(key, AES.MODE_CBC, i[-48:-32])
    f.write(t.decrypt(i[:-48]))
    f.write(i[-48:])
```

https://github.com/adysec/wechat_sqlite


DB Browser for SQLite - F:\VMs\weChatDB\msg\MSG0.db
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
新建数据库(N) 打开数据库(O) 写入更改(W) 倒退更改(R) 撤销(U) 打开项目(P) 保存项目(S) 附加
数据库结构(D) 浏览数据(B) 编辑杂注(R) 执行 SQL(X)
表(T): MSG
过滤任何列

	Sequence	StatusEx	FlagEx	Status	MsgServerSeq	MsgSequence	StrTalker	StrContent	Displ
	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤
1	1750217958000	0	0	2	0	0	lingdong	flag{root-46333405183428457667014104}	