

# wechatdb-siebel

---

端口扫描

访问 80

目录扫描

【我的进度到这了，其它找不到了，后来看群友发的 wp 看到是还有一段是在图片里，但是页面上并没有==...】

打开靶机直接给了 ip:10.133.98.30

```
* Seeding random number generator ...
* Seeding 256 bits and crediting
* Saving 256 bits of creditable seed for next boot
* Starting busybox syslog ...
* Starting busybox acpid ...
* Starting busybox crond ...
* /run/nginx: creating directory
* /run/nginx: correcting owner
* Starting nginx ...
* Starting busybox ntpd ...
* Starting sshd ...
* Starting local ...

== Virtual Machine Ready ==
IP Address: 10.133.98.30
QQ Group:660930334
Enjoy the game !!!
=====
```

端口扫描

10.133.98.30

ID	Host	Port	Protocol
1	10.133.98.30	22	
2	10.133.98.30	143	
3	10.133.98.30	80	

## 访问 80

▲ 不安全 10.133.98.30

**echo "Enjoy the game !!!"**

---

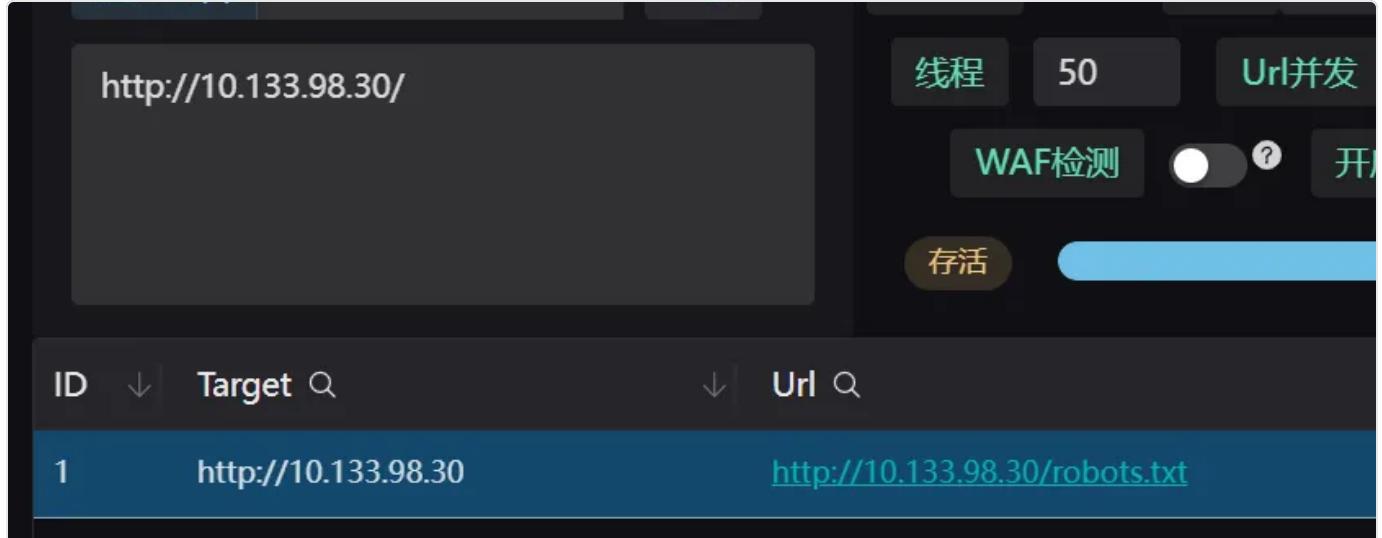
c:

```
248572329695973478362052182386835939741780647345295370384948765066264615802296090049815
18803643157663179774743410510655734763311765005052132120045336899544725507571957071600
90386005417706294793737099741836900763694331987804822291585926732616377830244486437626
343746089375428287395387898863530608997490939119682989457995477459194959047912600531203
350043180684424347179988873057991506761625275350195344920627652206158301633645063851914
041020965562368988326399931096094011388686864692459371985391446095940209482208687107282
464633340518342845766701410462637425834386891240008750978021095001641783628790655577247
420899204493495661801447201570098971419001438320104157507697184724381833340867103604721
02699572187850250950374788136798915912081043286101288981604811782615350141199334519304
35571689133202358200078549132019056426669961375970093095683168744327933776121734310182
480496674553633039532366325687060814567278922832199117170888096704981210784852454144575
13072740875601697161943037221934573531774258498914514353136657609930217500896418121168
5968251896719222083548713854836870552554007605129078062027799619840805034658967201
28842137190287628457401569072607796966676037243414037700424920214198274817913263498849
377952487257878279213177839237426230644979889588133465775583793170683180967230118928529
59981206826868898898281920910548518527710062483312775104203772108830571417599523697701
0639473655562183826537495742343998167395519036336021032361303551686519052382239461864
60615881670458647070208220375400209686965470237429765142158499926616499103381660308478
21782567182320225587379873493519910071500604554961665975367359390941637181034654345714
926761555214819025895212310772981499331595198872788941409926769504282914300207219186024
67108830737492247092636499917530826402928589794869294200385629317509406948837470268
522594283604795224671336014632806454725728058317936563252188781423271931103093
3932981276869584247773371232397986964608453661239422976346223178490810983167331510423
7504021685543680537749039667632119602726123947032016061853537287913331748697085083313
364536666319813968726925248834907259170258865155106278486963117508289087927072558836141
12961808726858599858998325603936821853866006160916937388256767627093516754446420453699
75094288397112653457646610978002679224876153589394319478994008888653876230053001366283
74823531783164261854044803391023889899365531411746805894404164037295813981845121460453
541734483237977850685616648974
```

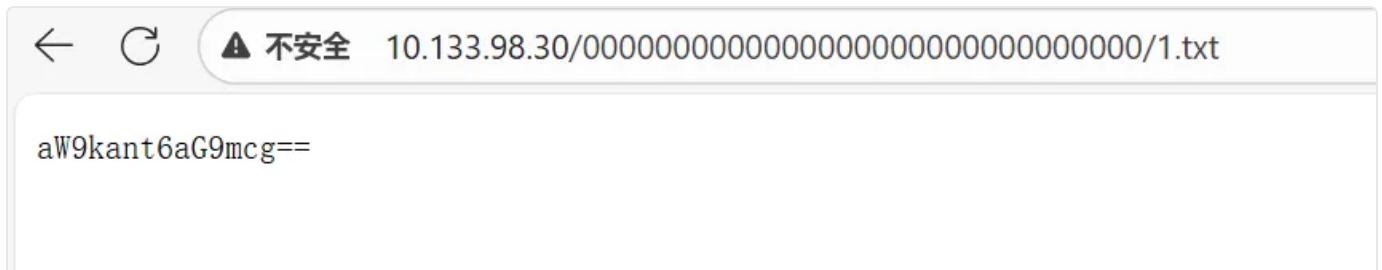
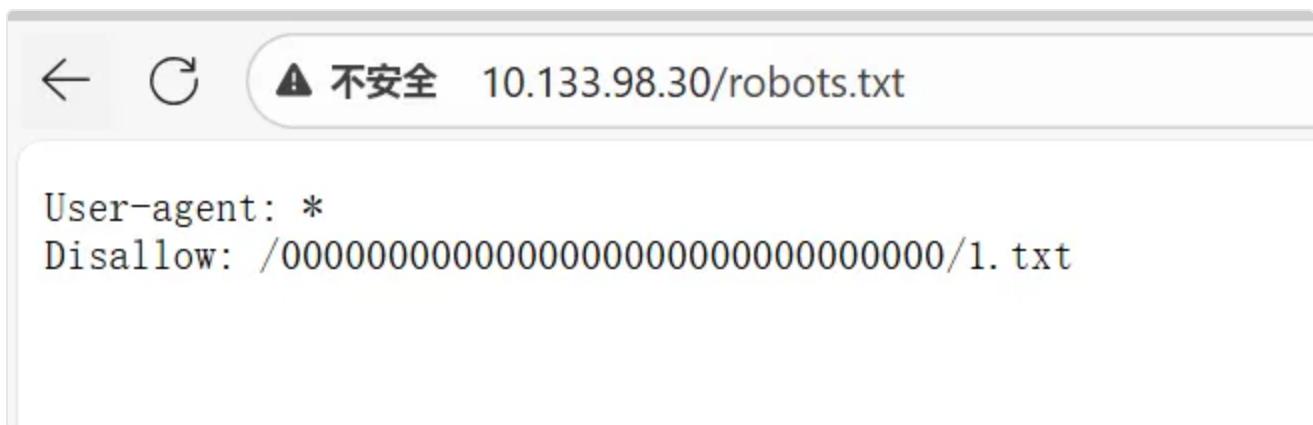
n:

```
695213614751622239008466798517335821785048478231076660010312963576116553599228196690026
107584614084172919726147096704484533476701096797379931545104638371750128092345556813521
```

目录扫描



访问 ip/robots.txt



## base64 解密

aW9kant6aG9mcg==

iodj{zhofr

像 flag 的一部分，应该是有偏移，上工具看一下

Base/Rot 字密1 字密2 字密3 字密4 编码转换 带key解密 多key解密 进制转  
输入 ↓ (字: 10) 密钥key/str/url:   
iodj{zhofr

输出 ↓     
mode1 #0: iodj{zhofr  
mode1 #1: hnciygneq  
mode1 #2: gmbhfxfmdp  
mode1 #3: flag{welco  
mode1 #4: ekzfvdkbn  
mode1 #5: djye{ucjam

得到道一个看着正常的： flag{welco

返回 首页， 页面给出的是 c,n,e

甩 ai 写个脚本或者往上找一个跑一下， 得到 me:wlc0mE@

```
D:\Exclusion_items\密码\脚本>python rsa.py
== 开始RSA解密流程 ==
正在分解模数n (弱RSA, 分解速度较快) ...
✓ 分解成功 (p和q过长, 仅显示前20位) :
    p = 83379470779780213174...
    q = 83379470779780213174...
✓ 欧拉函数φ(n) (仅显示前30位) : 695213614751622239008466798517...
✓ 私钥d (仅显示前30位) : 562847539330551053291274239933...
正在解密密文c...
✓ 密文解密完成, 明文整数形式 (仅显示前50位) : 516605280739385691424064...
✓ 明文转换成功: me:wlc0mE@
== RSA解密流程结束 ==

=====
国旗 鞠 为: me:wlc0mE@
=====
```

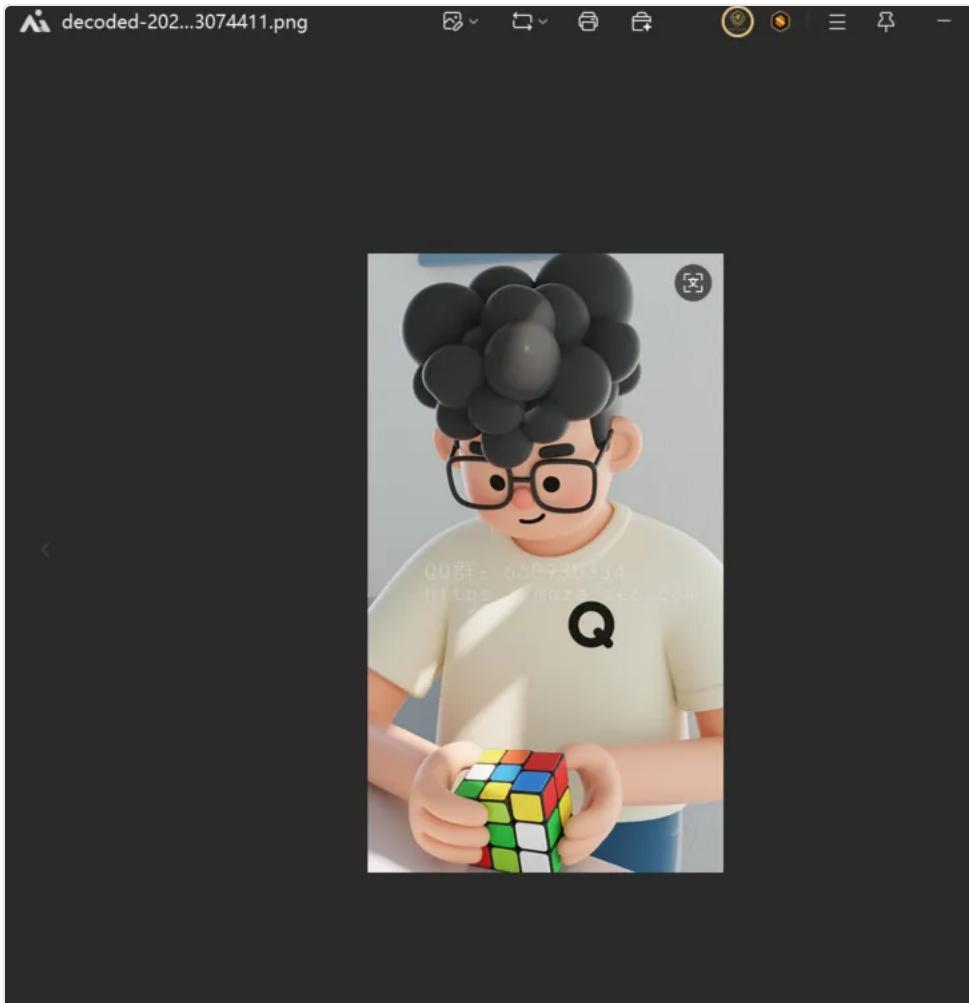
【我的进度到这了，其它找不到了，后来看群友发的 wp 看到是还有一段 是在 图片里，但是页面上并没有==】

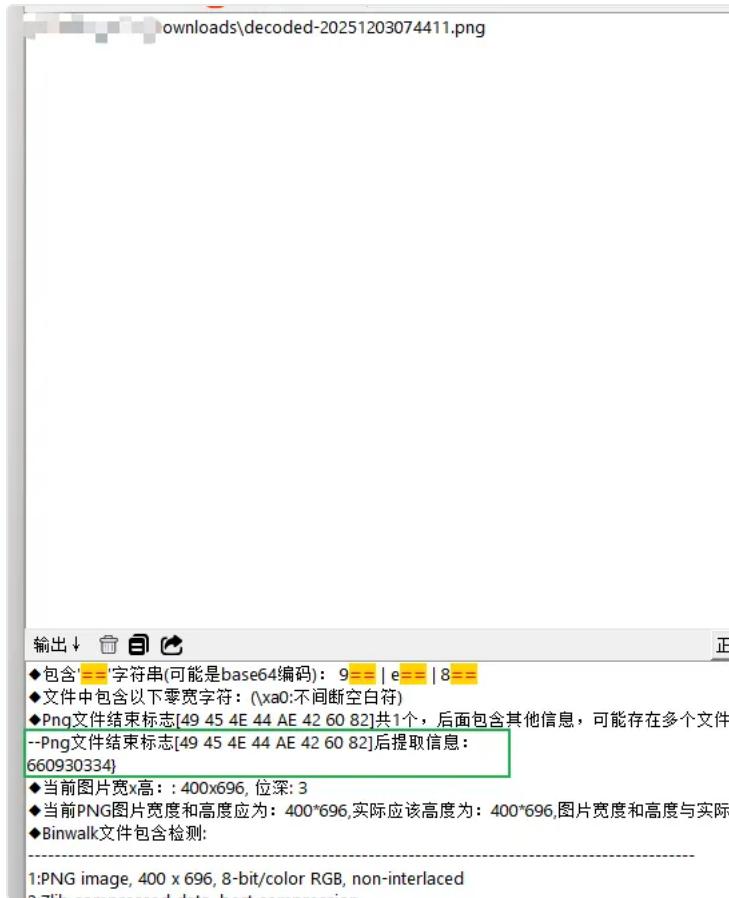
```

▲ 不安全 view-source:10.133.98.30

17     border-bottom: 2px solid #0779e4;
18     padding-bottom: 10px;
19   }
20   .data-item {
21     margin-bottom: 25px;
22     padding: 15px;
23     background-color: #f9f9f9;
24     border-radius: 5px;
25     border-left: 4px solid #0779e4;
26   }
27   .data-label {
28     font-weight: bold;
29     color: #333;
30     margin-bottom: 8px;
31     font-size: 1.1em;
32   }
33   .data-value {
34     font-family: 'Courier New', monospace;
35     word-break: break-all;
36     white-space: pre-wrap;
37     background-color: #fff;
38     padding: 12px;
39     border-radius: 4px;
40     border: 1px solid #ddd;
41   }
42   .test{
43     background-image: url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAZAAAAAK4CAIAAADVyjxdAAAACXBIVXMAAsTAAALEwEAmpwYAAALc21UWHRYTUw6Y29tLmFkb2
44   }
45   
```

- 用在线工具（如 [Base64 解码](#)）解码 Base64 字符串（去掉 `data:image/png;base64,` 前缀）；
- 解码后下载「二进制文件」，重命名为 `.png` 即可。





Png文件结束标志 [49 45 4E 44 AE 42 60 82]后包含其他信息，提取到的片段为 660930334}。

OK 拼起来 flag{welcome:wlc0mE@660930334}

```
▼ Bash |  
1 ssh welcome@10.133.98.30
```

然后输入密码 wlc0mE@660930334

```
1  [root@kali] -[/home/kali]
2  [root@kali ~]# ssh welcome@10.133.98.30
3  welcome@10.133.98.30's password:
4  =====
5  Welcome!!!
6  QQ Group:660930334
7  =====
8  lingdong:~$ ls
9  tip.txt      user.txt      wechat_files
10 lingdong:~$ cat user.txt
11 flag{user-415621D5297F8F4BE138A5BB03}lingdong:~$
```

将文件打包拿出来

```
scp welcome@10.133.98.30:~/wechat_files.tar.gz .
```

用 ai 或者网上找一个微信PC端数据库文件解密脚本

```
1  from Crypto.Cipher import AES
2  import hashlib
3  import hmac
4  import ctypes
5  import sys
6
7  # 微信数据库加密固定参数（无需修改）
8  SQLITE_FILE_HEADER = bytes('SQLite format 3', encoding='ASCII') + b'\x00'
    # 修复：补全SQLite文件头
9  IV_SIZE = 16
10 HMAC_SHA1_SIZE = 20
11 KEY_SIZE = 32
12 DEFAULT_PAGESIZE = 4096
13 DEFAULT_ITER = 64000
14
15
16 def decrypt_wechat_db():
17     """
18         微信数据库解密脚本（修复语法错误+优化交互）
19         支持输入密钥和文件路径，自动处理解密并覆盖原文件（或可修改为输出新文件）
20     """
21     try:
22         # 1. 交互式输入参数（无需手动改代码）
23         input_pass = input('请输入十六进制密钥（不含空格/0x前缀）：').strip()
24         input_dir = input('请输入数据库文件路径（如 ./MSG0.db）：').strip()
25
26         # 2. 处理密钥（去除空格，转换为bytes）
27         try:
28             password = bytes.fromhex(input_pass.replace(' ', ''))
```

```
29         except ValueError:
30             print('✖ 密钥格式错误！请输入纯十六进制字符串（0-9、a-f/A-F）')
31             return
```

```
32
33         # 3. 读取加密数据库文件
34         try:
35             with open(input_dir, 'rb') as f:
36                 blist = f.read()
37         except FileNotFoundError:
38             print(f'✖ 未找到文件：{input_dir}（请检查路径是否正确）')
39             return
40         except PermissionError:
41             print(f'✖ 无权限读取文件：{input_dir}（请以管理员身份运行）')
42             return
```

```
43
44         print(f'✅ 成功读取文件，文件大小：{len(blist)} 字节')
45
46         # 4. 提取盐值并生成AES密钥（微信标准PBKDF2算法）
```

```

47     salt = blist[:16]
48     key = hashlib.pbkdf2_hmac(
49         'sha1',
50         password,
51         salt,
52         DEFAULT_ITER,
53         KEY_SIZE
54     )
55
56     # 5. 校验密码（通过第一页数据的HMAC校验）
57     first_page = blist[16:DEFAULT_PAGESIZE] # 第一页数据（跳过前16字节盐
58     值)
59     if len(first_page) < 48:
60         print('✖ 文件损坏! 第一页数据不完整')
61         return
62
63     # 生成MAC校验密钥
64     mac_salt = bytes([x ^ 58 for x in salt])
65     mac_key = hashlib.pbkdf2_hmac('sha1', key, mac_salt, 2, KEY_SIZE)
66
67     # 计算HMAC并校验
68     hash_mac = hmac.new(mac_key, digestmod='sha1')
69     hash_mac.update(first_page[:-32]) # 第一页除末尾32字节外的内容
70     hash_mac.update(bytes(ctypes.c_int(1))) # 页数标识（第一页为1）
71     if hash_mac.digest() != first_page[-32:-12]: # 末尾32-12=20字节为
72         HMAC-SHA1值
73         print('✖ 密码错误! HMAC校验失败')
74         return
75     print('✓ 密码校验通过，开始解密...')

76     # 6. 分割剩余数据为分页（按SQLite页大小4096分割）
77     remaining_pages = [
78         blist[i:i + DEFAULT_PAGESIZE]
79         for i in range(DEFAULT_PAGESIZE, len(blist), DEFAULT_PAGESIZE)
80     ]
81
82     # 7. 解密并写入文件（覆盖原文件，如需保留原文件可修改输出路径）
83     with open(input_dir, 'wb') as f:
84         # 写入SQLite标准文件头（解密后数据库需识别为SQLite格式）
85         f.write(SQLITE_FILE_HEADER)
86
87         # 解密第一页数据
88         iv = first_page[-48:-32] # 第一页末尾48-32=16字节为IV
89         cipher = AES.new(key, AES.MODE_CBC, iv)
90         decrypted_first = cipher.decrypt(first_page[:-48]) # 第一页除
未尾48字节外的内容
         f.write(decrypted_first)

```

```

91             f.write(first_page[-48:]) # 保留IV和HMAC数据（不影响数据库读取）
92
93         # 解密剩余分页
94         for page in remaining_pages:
95             if len(page) < 48:
96                 print(f'⚠ 跳过不完整分页（大小: {len(page)} 字节）')
97                 continue
98             iv = page[-48:-32]
99             cipher = AES.new(key, AES.MODE_CBC, iv)
100            decrypted_page = cipher.decrypt(page[:-48])
101            f.write(decrypted_page)
102            f.write(page[-48:])
103
104        print(f'✅ 解密成功！文件已保存为: {input_dir}')
105
106    except Exception as e:
107        print(f'❗️ 解密失败: {str(e)}')
108        return
109
110
111 if __name__ == "__main__":
112     print('=' * 50)
113     print('微信数据库解密工具（修复版）')
114     print('=' * 50)
115     decrypt_wechat_db()

```

```

D:\Exclusion_items\密码\脚本\wechat_files>python wechatdb.py
=====
微信数据库解密工具（修复版）
=====
请输入十六进制密钥（不含空格/0x前缀）: c22ce55044354439b22d75a1e1e4be286bc480cde0f34583bb490fe686b56061
请输入数据库文件路径（如 ./MSG0.db）: D:\Exclusion_items\密码\脚本\wechat_files\lingdong\msg\MSG0.db
✅ 成功读取文件，文件大小: 52428800 字节
✅ 密码校验通过，开始解密...
✅ 解密成功！文件已保存为: D:\Exclusion_items\密码\脚本\wechat_files\lingdong\msg\MSG0.db

D:\Exclusion_items\密码\脚本\wechat_files>

```

使用脚本还原微信数据库 `MSG0.db` 的明文数据，然后用数据库工具打开

MSG0) - 表 - Navicat Premium

The screenshot shows the Navicat Premium interface for a database named 'MSG0'. The main window title is 'MSG0) - 表 - Navicat Premium'. The menu bar includes '查看' (View), '表' (Table), '收藏夹' (Favorites), '工具' (Tools), '窗口' (Windows), and '帮助' (Help). Below the menu is a toolbar with icons for '建查询' (Build Query), '表' (Table), '视图' (View), '索引' (Index), '触发器' (Trigger), '用户' (User), '查询' (Query), '备份' (Backup), '自动运行' (Automatic Run), '模型' (Model), and '图表' (Chart). A navigation pane on the left lists objects: 'Info', 'Trans', and 'Name2ID'. The central area displays a table named 'MSG' with the following data:

对象	MSG @main (MSG0) - ...	DBInfo @main (MSG0)...	MSGTrans @main (MS...)	Name2ID @ma...
提交	回滚	文本	筛选	排序
MsgServerSeq	MsgSequence	StrTalker	StrContent	DisplayCon...
0	0	lingdong	flag{root-46333405183428457667014104}	

flag{root-46333405183428457667014104}