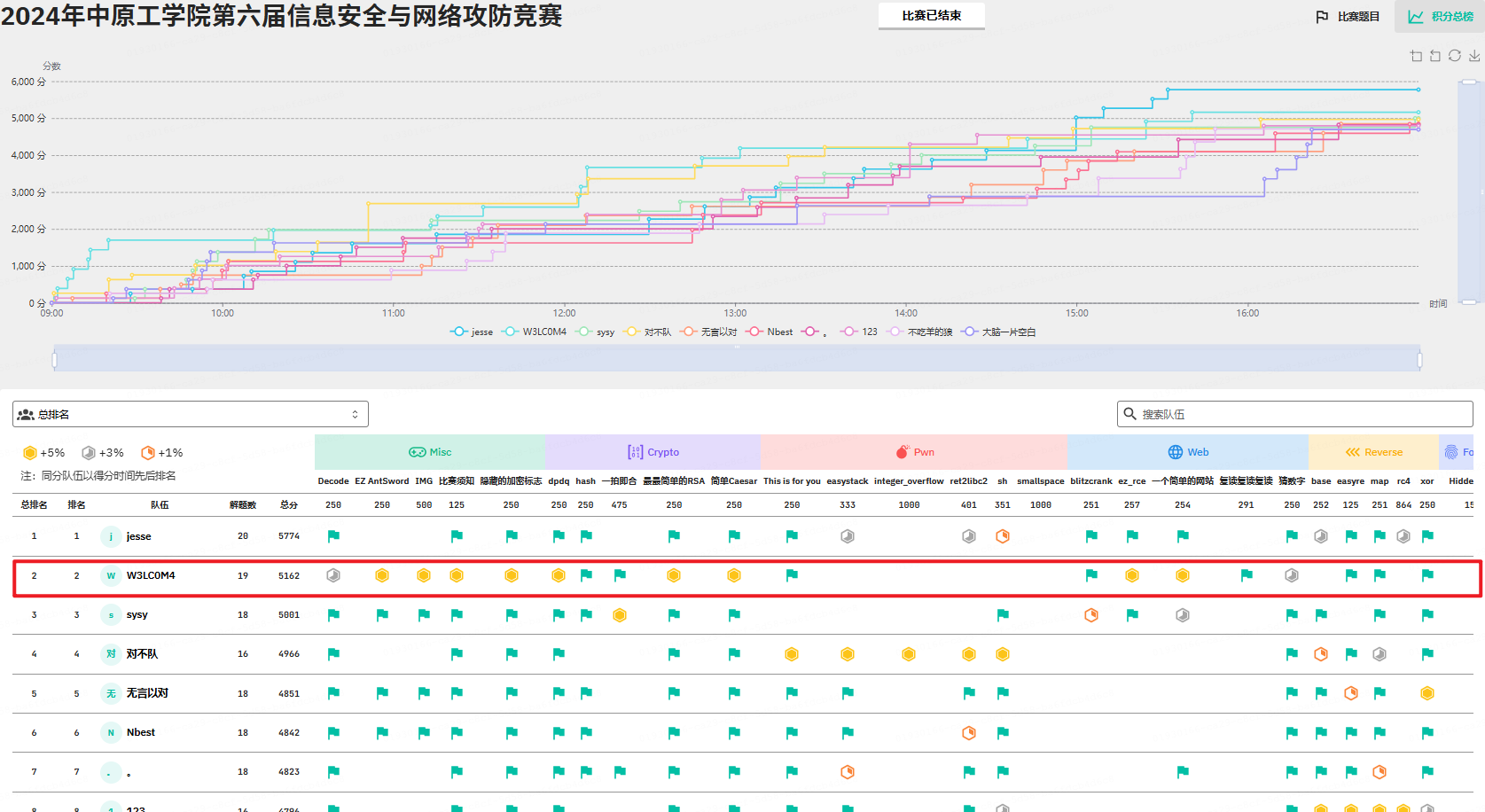
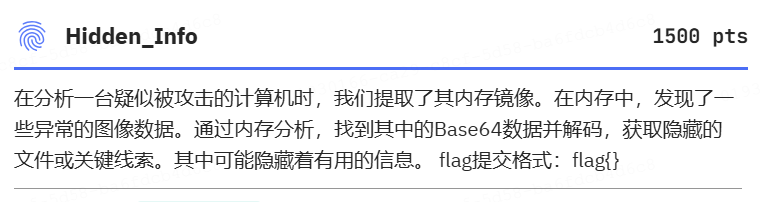
ZUT第六届信息安全与网络攻防竞赛WP

2024-12-24

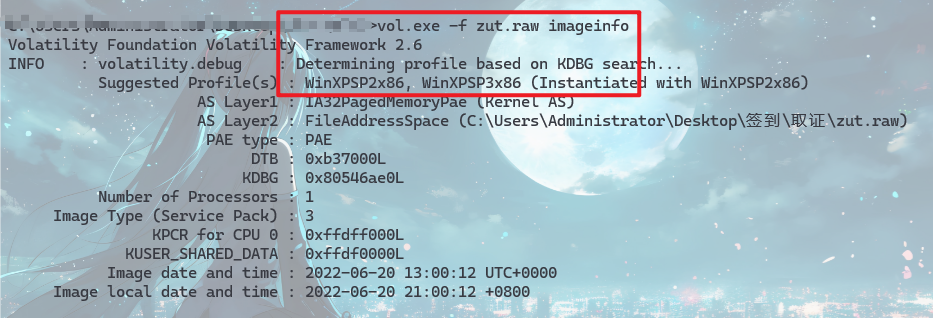
# 第六届信息安全与网络攻防竞赛WP



## Hidden\_Info



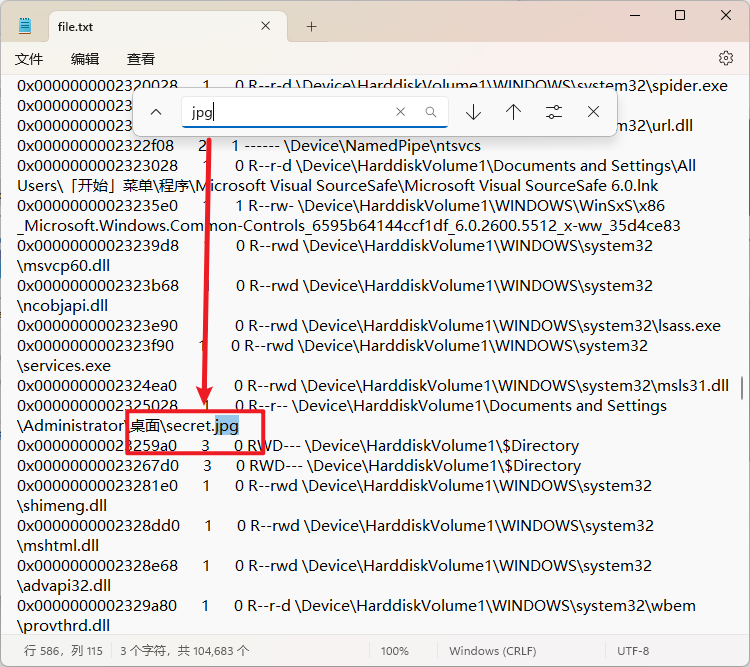
首先查看镜像信息

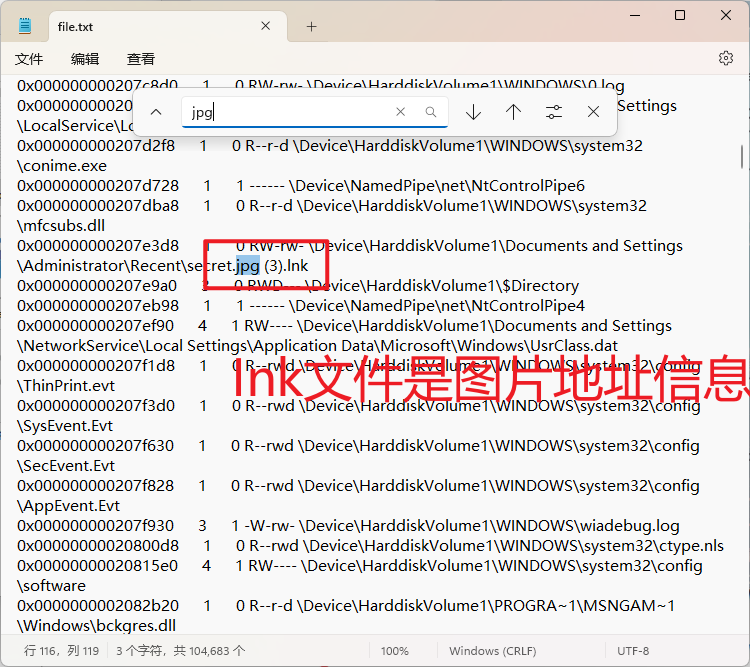


接着查看里边的文件信息并重定向保存出来

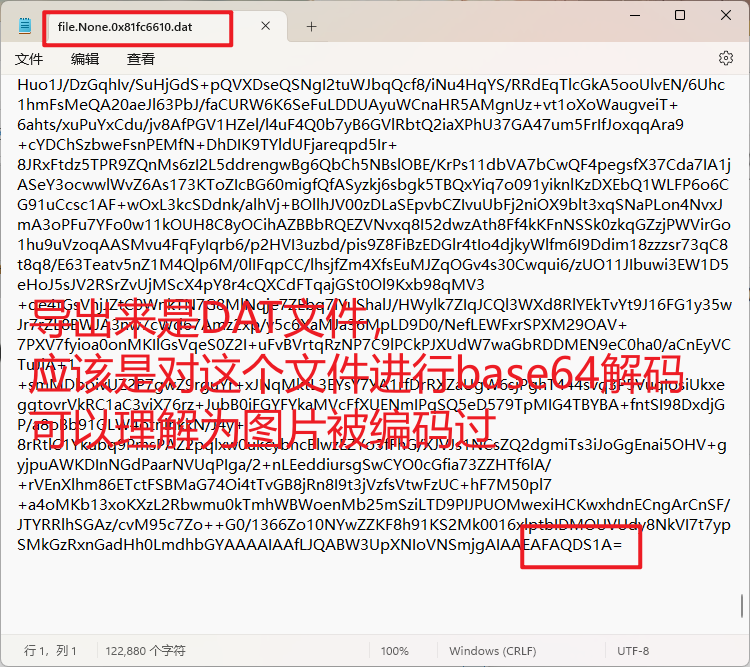
vol.exe -f zut.raw --profile=WinXPSP2x86 filescan > file.txt

根据提示信息查找图片





vol.exe -f zut.raw --profile=WinXPSP2x86 dumpfiles -Q 0x0000000002325028 -D ./  
  
dumpfiles 导出指定的文件内容到本地  
-Q 图⽚⾥⾯左边的16进制 应该是对应编号  
-D 保存路径



注意：不能用在线工具解码，可能会因为字符限制长度导致解码不完全

还是脚本比较稳妥一些



AI生成脚本进行解码

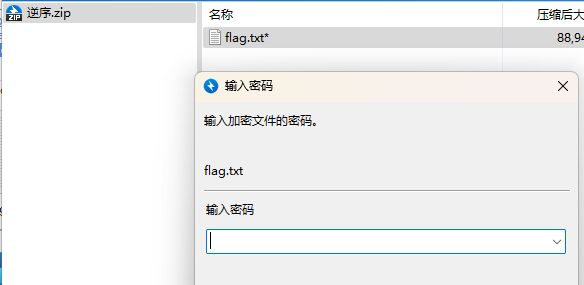
import base64  
  
# 打开并读取 1.dat 文件中的 base64 编码数据  
with open("1.dat", "rb") as input\_file:  
 encoded\_data = input\_file.read() # 读取文件内容  
  
# 对数据进行 base64 解码  
decoded\_data = base64.b64decode(encoded\_data)  
  
# 将解码后的数据保存为 flag.txt  
with open("flag.txt", "wb") as output\_file:  
 output\_file.write(decoded\_data) # 写入解码后的数据  
  
print("Base64 解码完成，结果已保存为 flag.txt")



脚本逆序回去

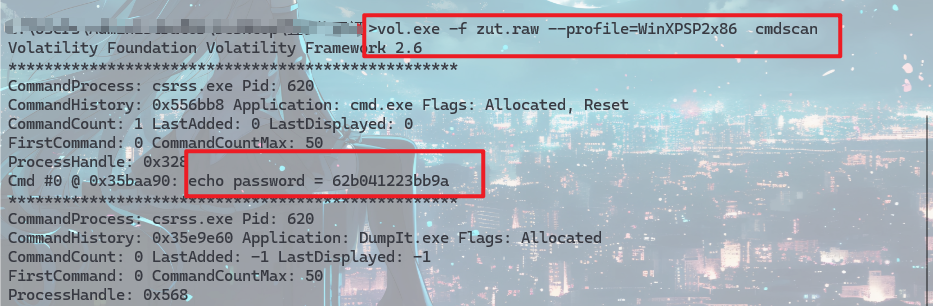
# 打开文件 flag.txt 并读取其内容  
with open("flag.txt", "rb") as input\_file:  
 data = input\_file.read() # 读取文件内容，返回字节数据  
  
# 反转字节数据  
reversed\_data = data[::-1]  
  
# 将反转后的数据保存回 flag.txt  
with open("flag.txt", "wb") as output\_file:  
 output\_file.write(reversed\_data)  
  
print("文件内容已逆序并保存为 flag.txt")

逆序完改为zip打开后发现要密码



查看历史命令看是否有密码信息

cmdscan获取密码



cmdscan  获取历史命令  
vol.exe -f zut.raw --profile=WinXPSP2x86 cmdscan



看的出来flag.txt中为像素坐标

转换为图片即可



将获得的二维码进行扫描

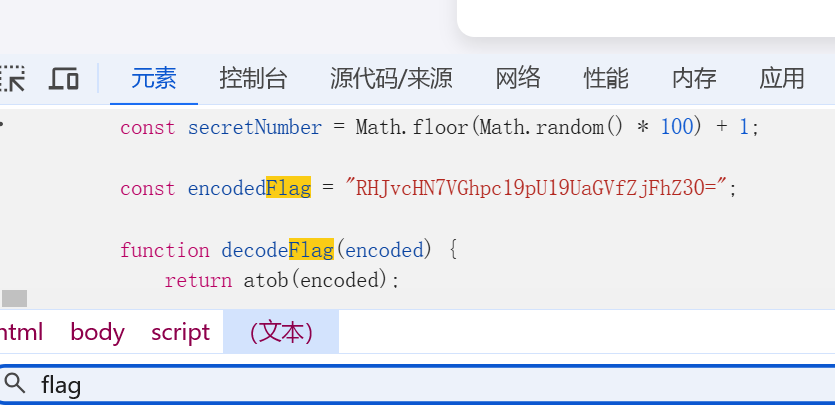
即可获得flag



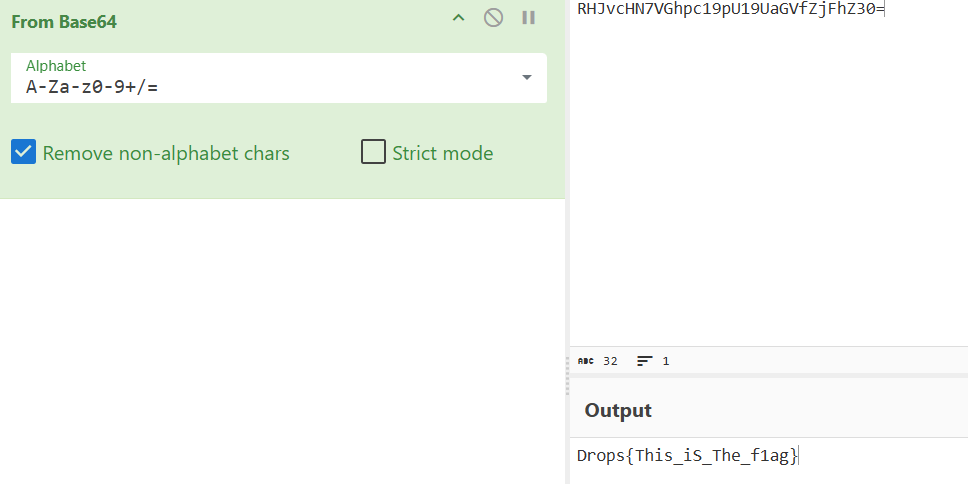
## Web

### 猜数字

查看源码



解密即可



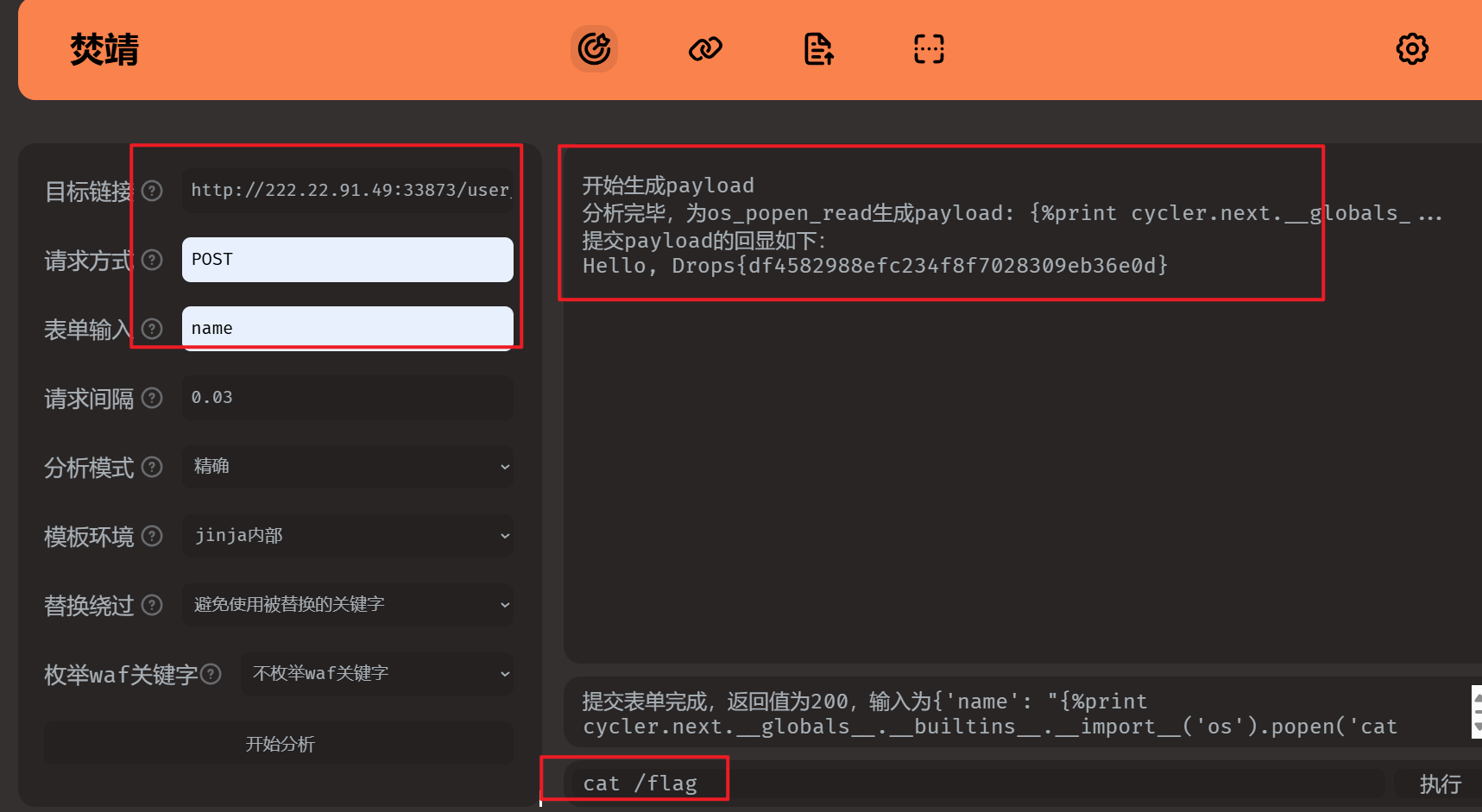
### 复读复读复读

进行输入



应该是SSTI

直接上fenjing

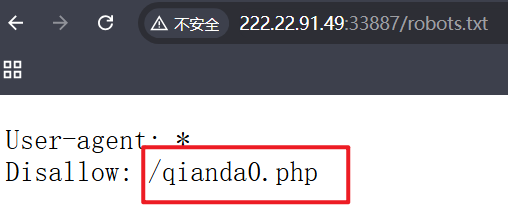


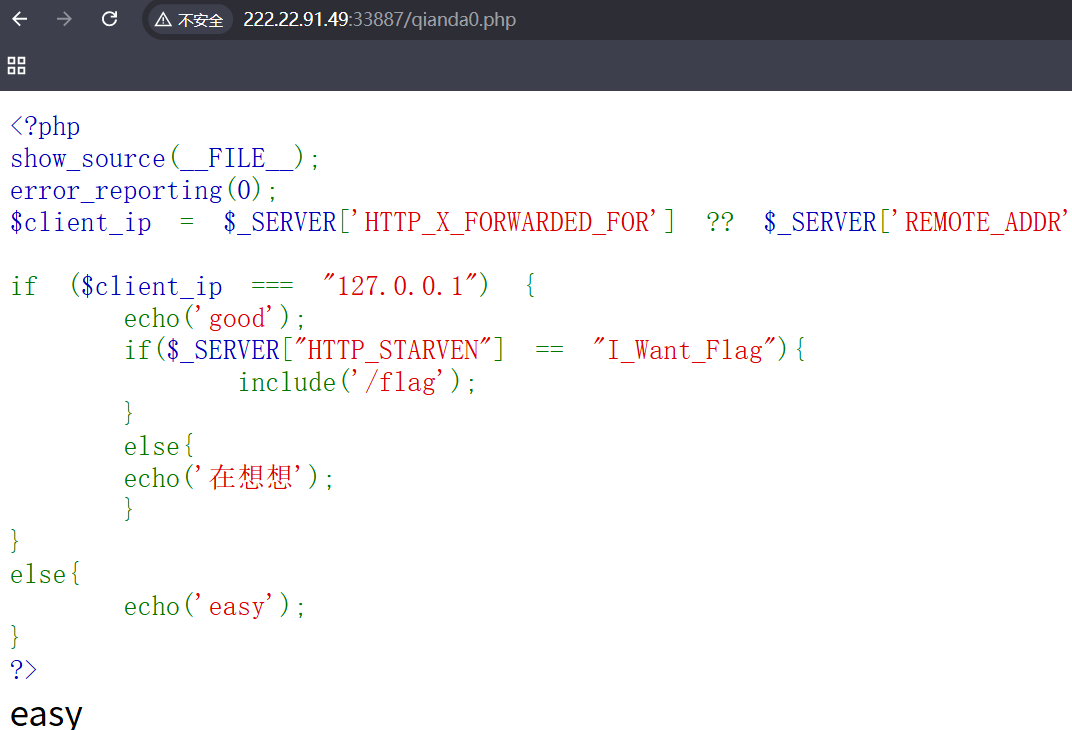
### blitzcrank



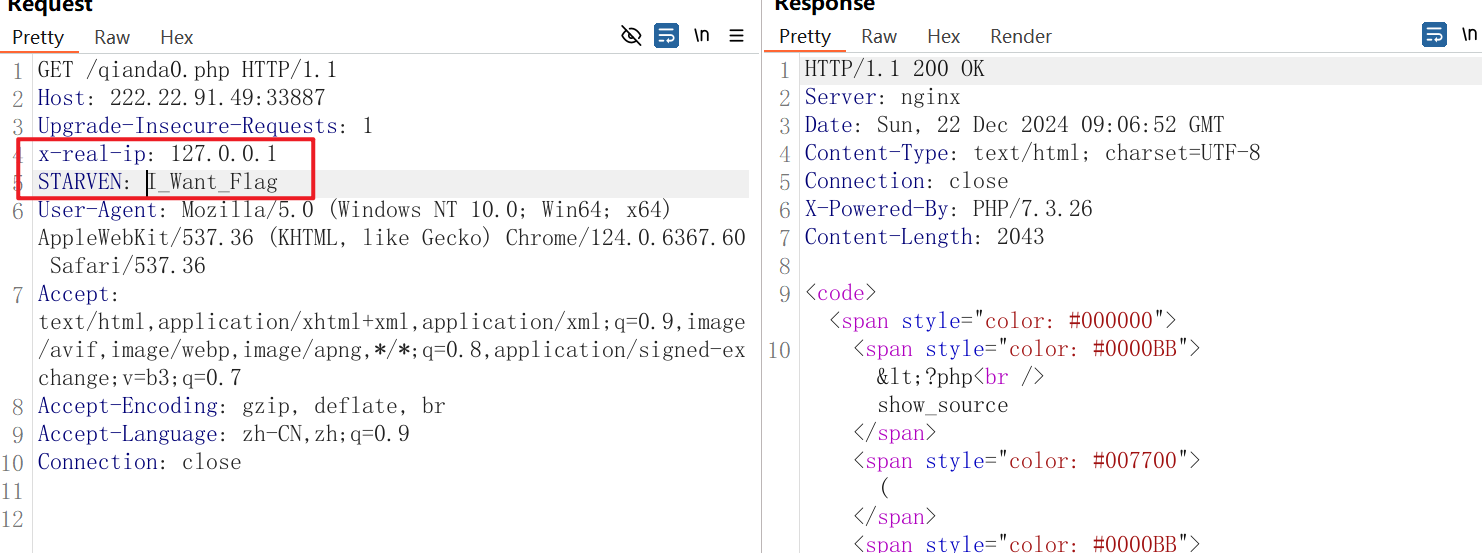
扫描目录纯在敏感文件

访问robots





代码分析，对应修改文件头即可



### ez\_rce

if (  
 sha1((string) $\_POST["\_\_2024.zut.ctf"]) == md5("QLTHNDT") &&  
 (string) $\_POST["\_\_2024.zut.ctf"] != "QLTHNDT" &&  
 is\_numeric(intval($\_POST["\_\_2024.zut.ctf"]))  
) {

\_\_2024.zut.ctf 双下划线传参

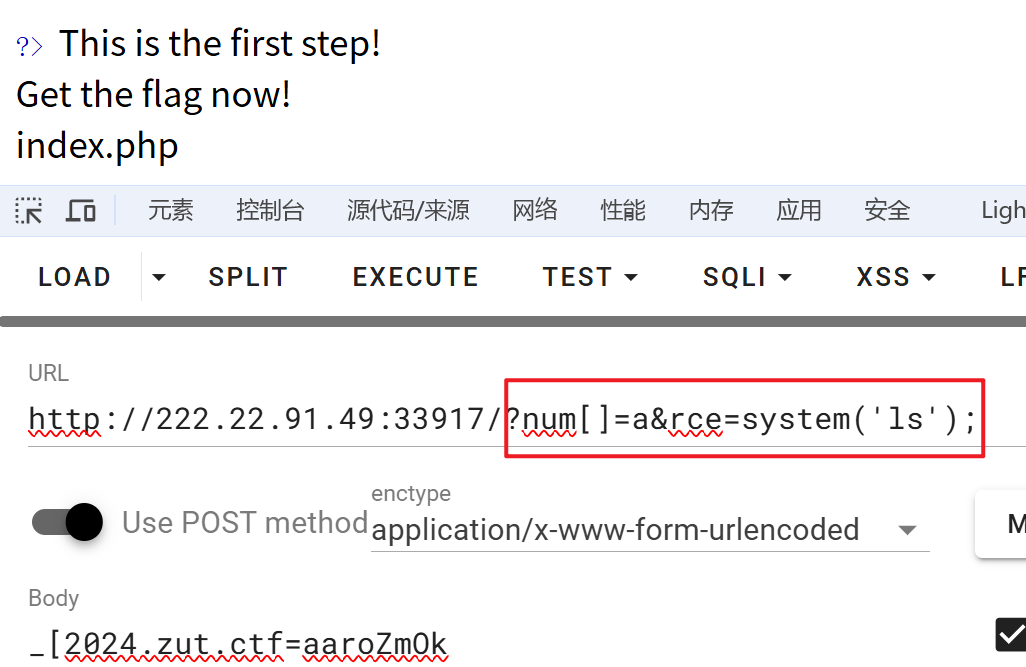
QLTHNDT的md5是0e字符串

找一个sha10e字符串绕过即可

\_[2024.zut.ctf=aaroZmOk







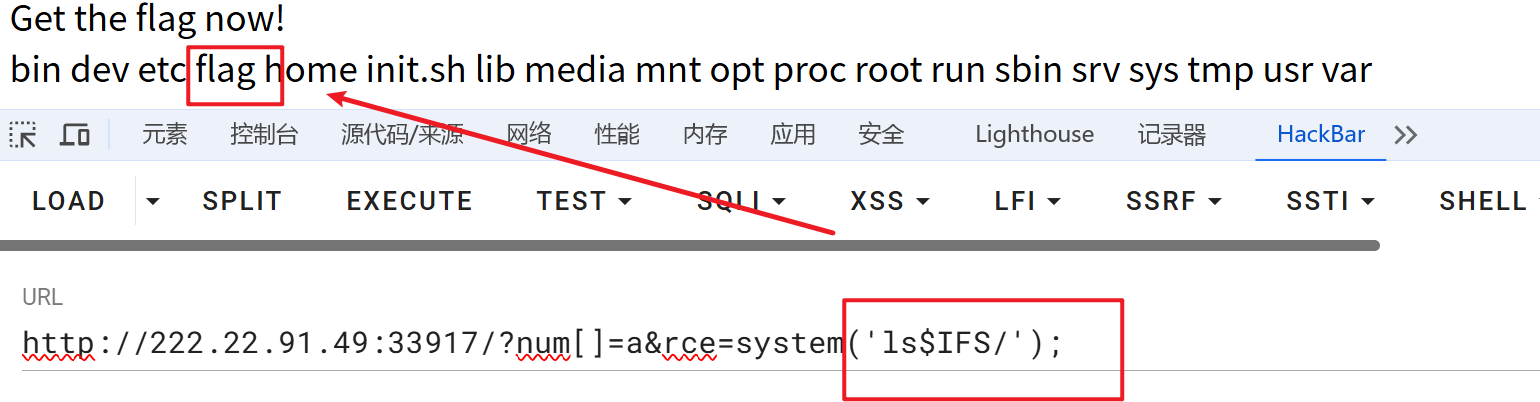
if(preg\_match("/cat| |flag/i",$rce)){  
 die("no no no!");  
 }else{  
 eval($rce);  
 }

过滤cat 空格 以及flag



查看根目录无结果

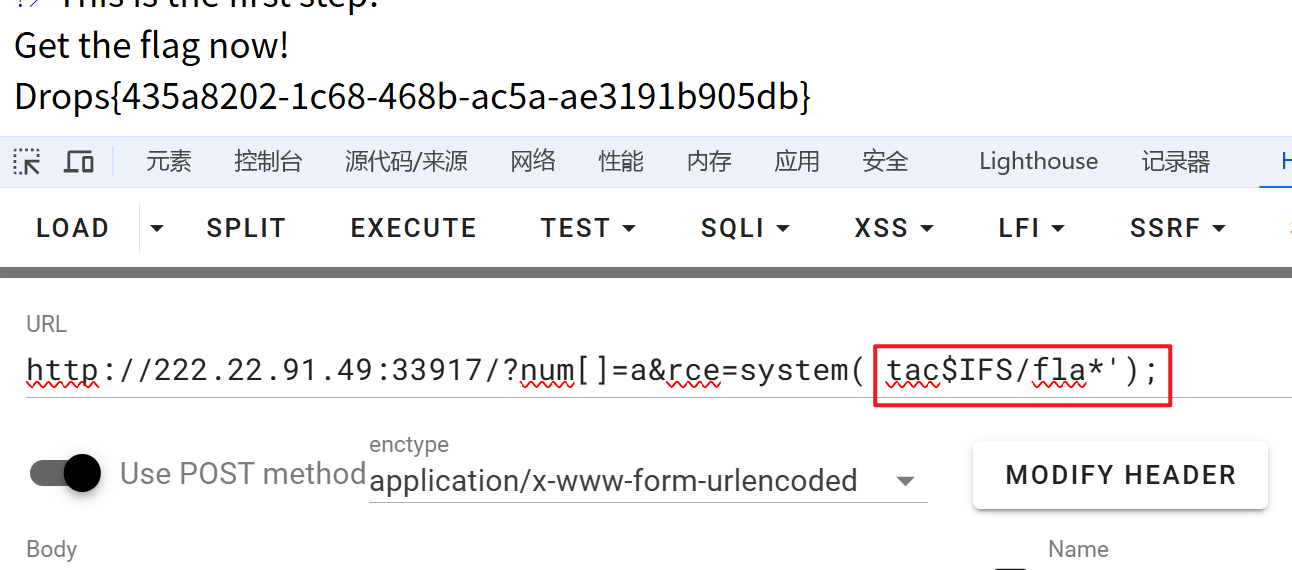
有空格



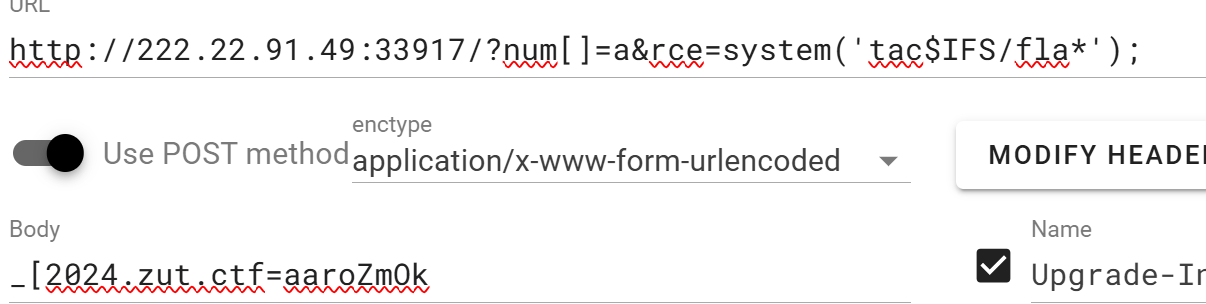
tac读取

&IFS代替空格

通配符绕过匹配



绕过成功

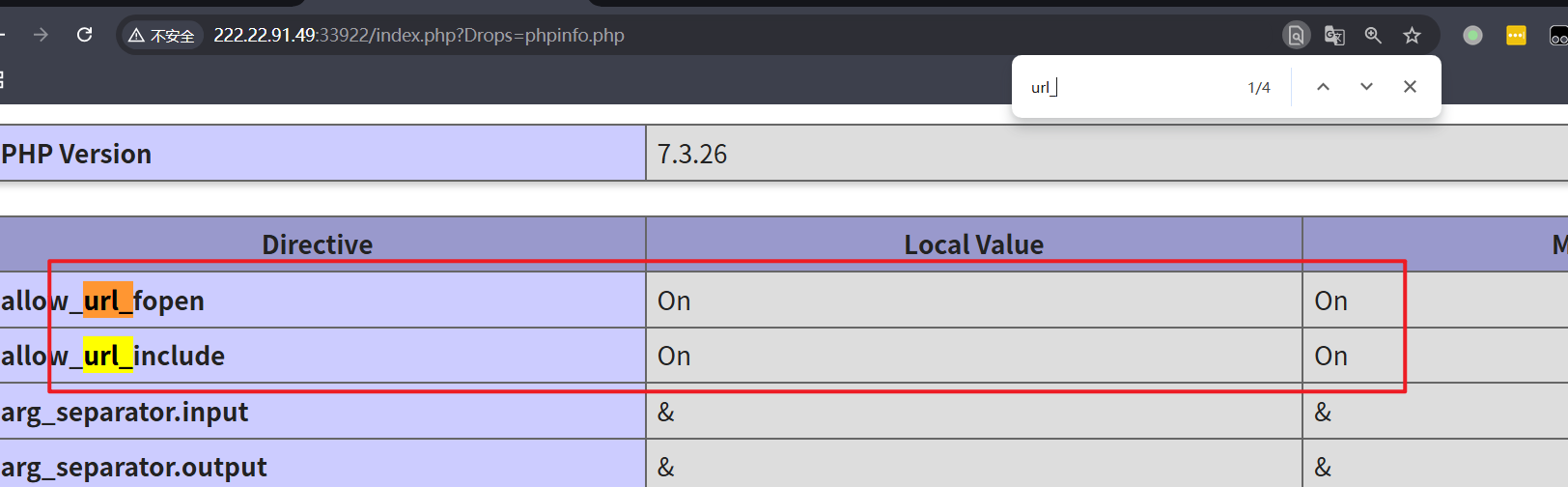


### 一个简单的网站



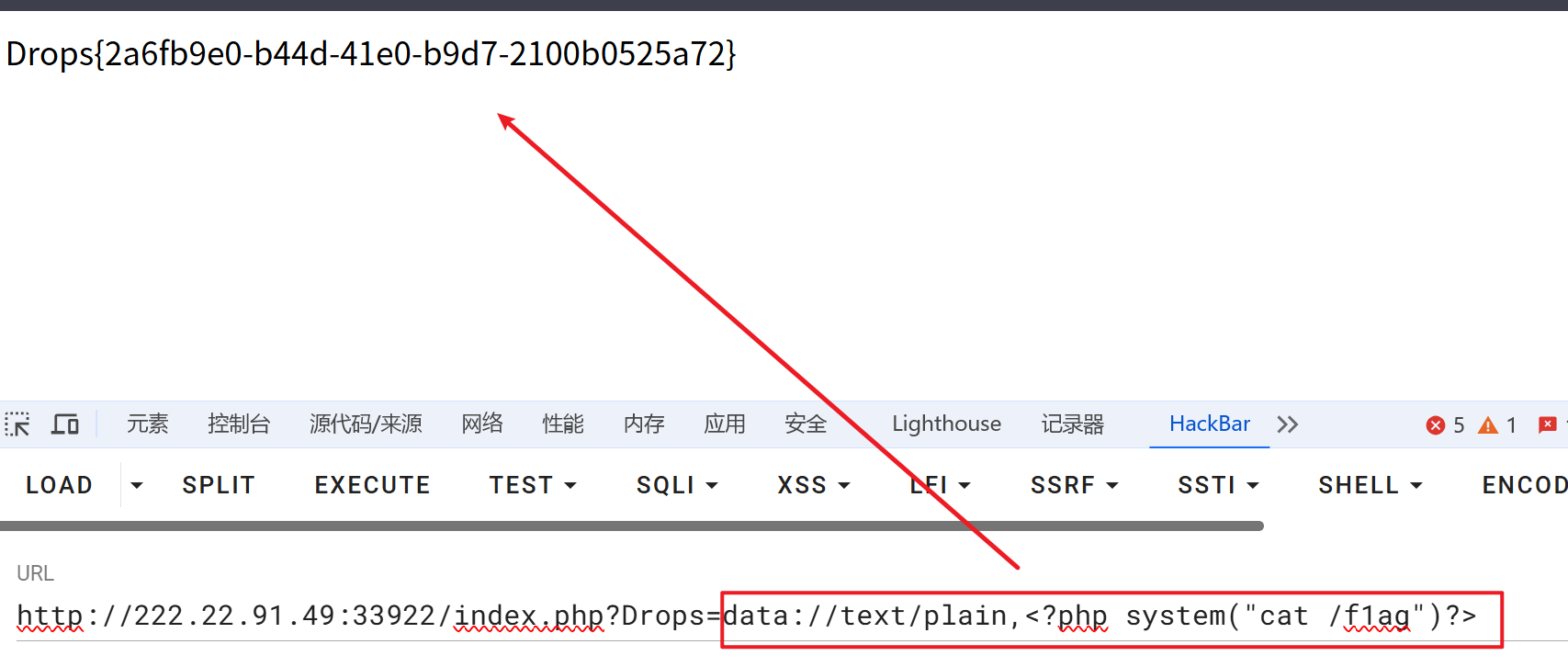
提示phpinfo

进去查看



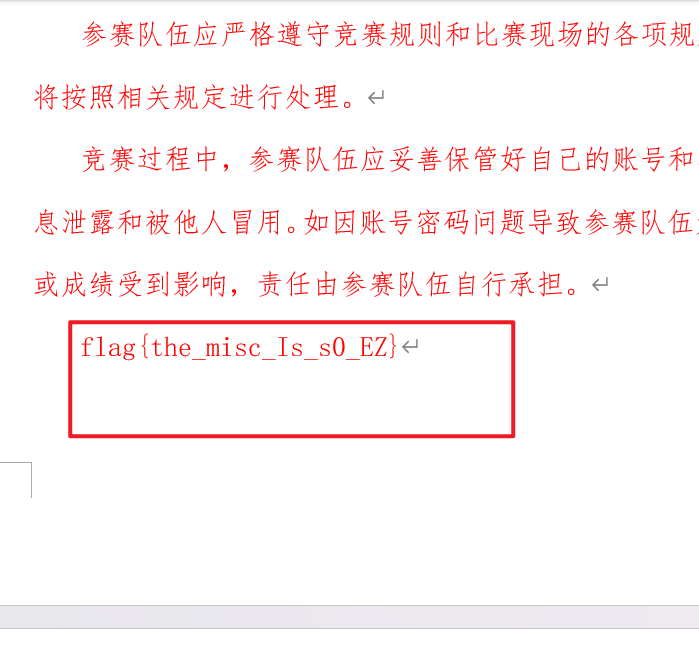
都是开的，php伪协议执行

data协议读取flag即可



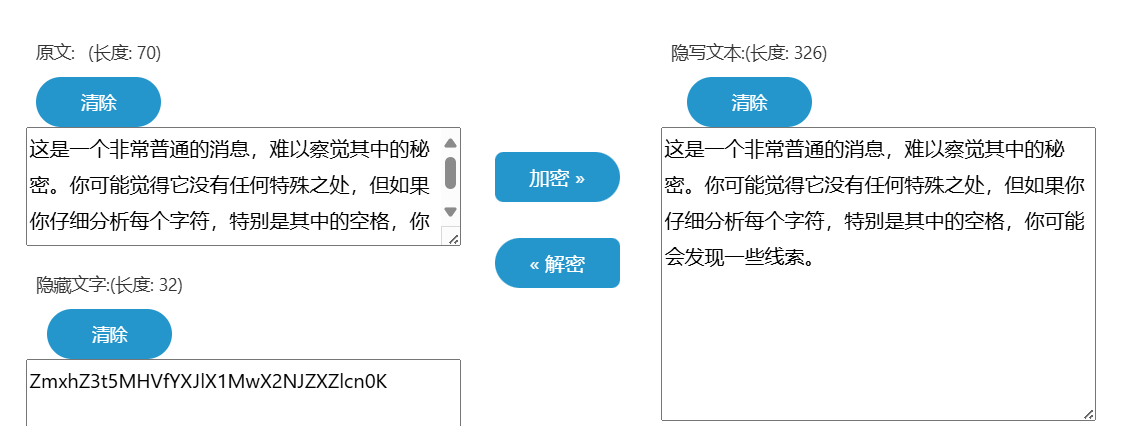
## Misc

### 比赛须知

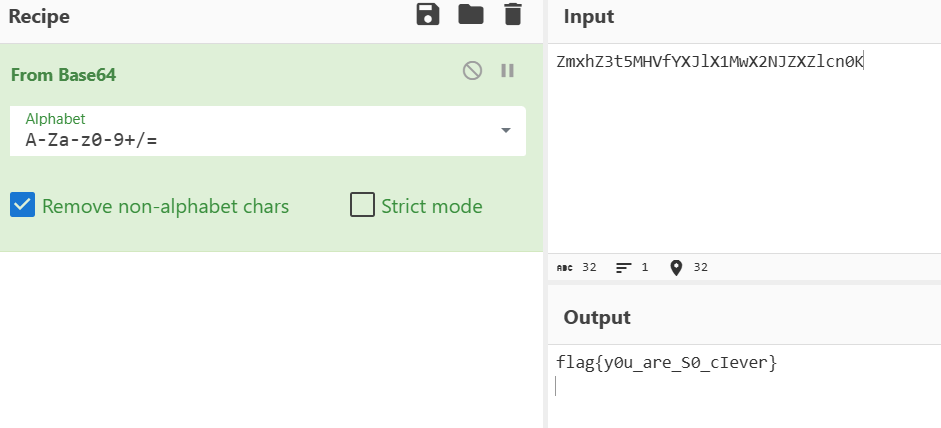


### 隐藏的加密标志

一眼零宽

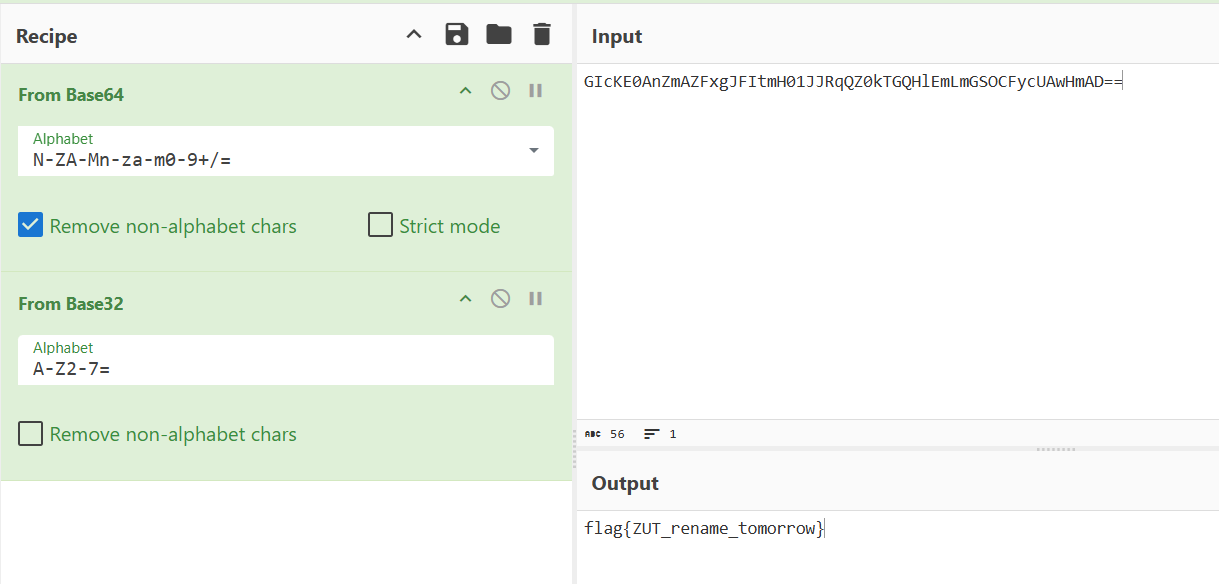


cyberchef直接

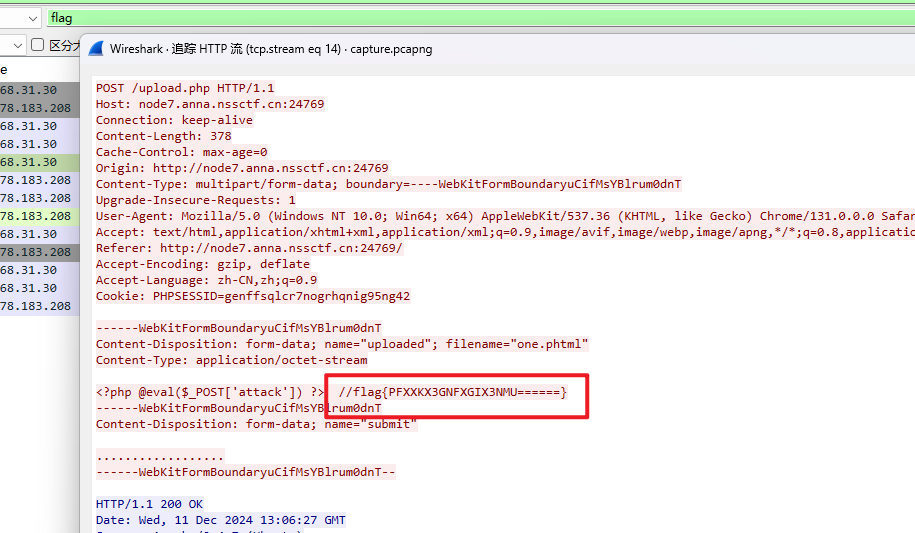


### Decode

直接出

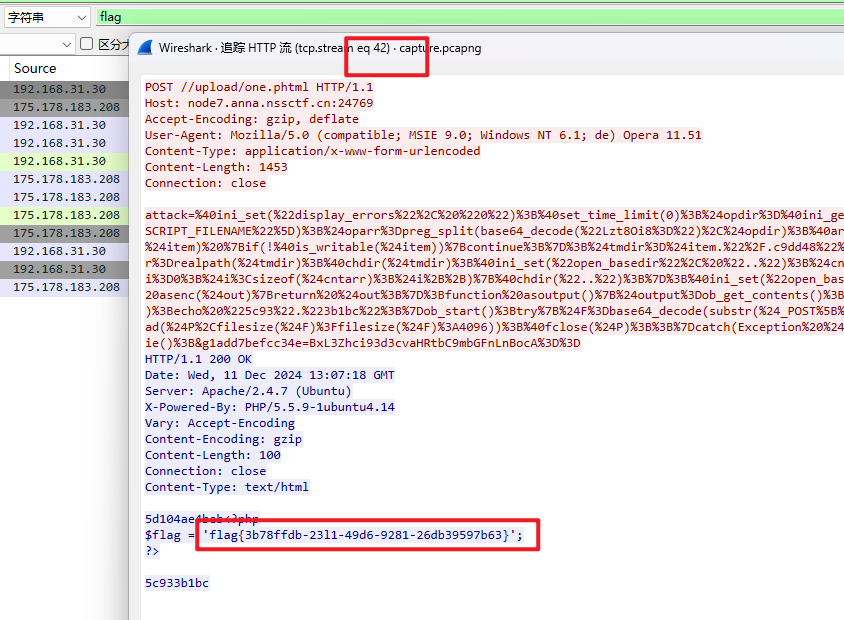


### EZ AntSword



找到flag 是假的

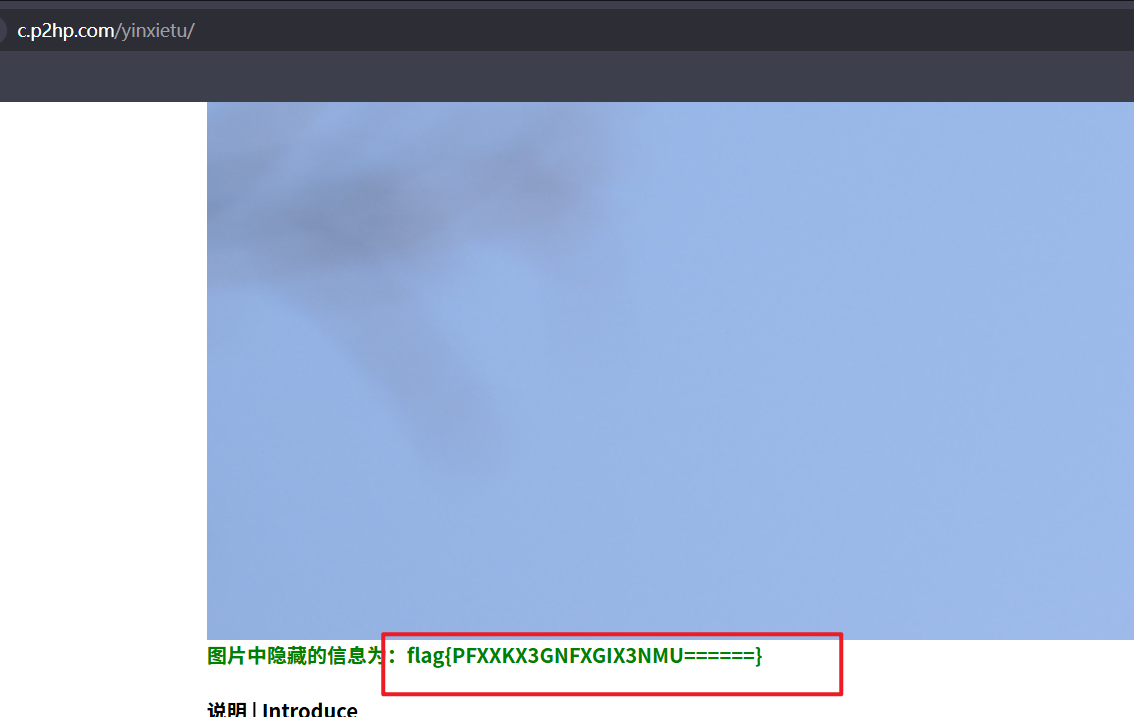
追踪流查看



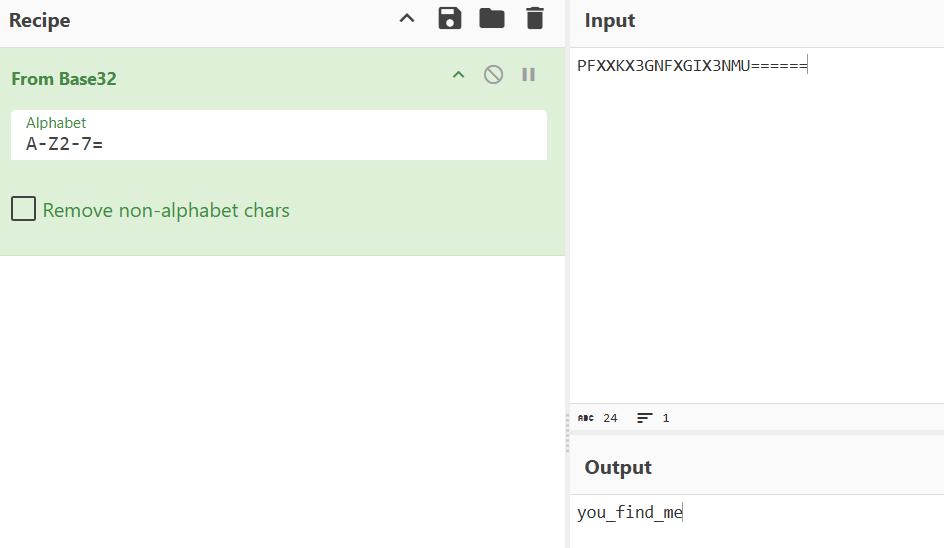
在后续的第42个流中找到flag

### IMG

常规工具都尝试了，不行，试试在线的



解码后flag包裹即可



## Crypto

### 最最简单的RSA

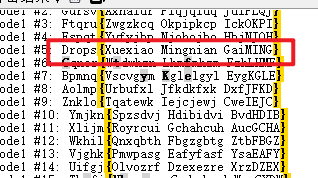
简单的RSA

上脚本



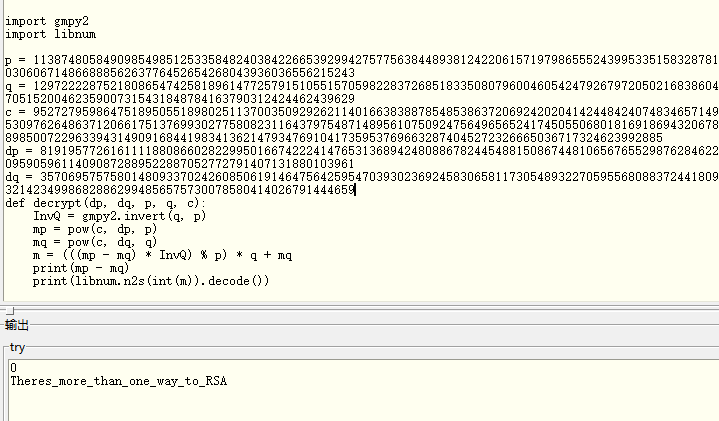
### 简单Caesar

直接凯撒



### dpdq

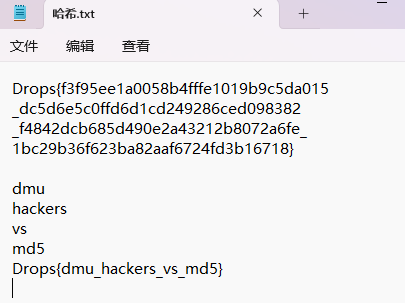
直接上脚本



#### exp

import gmpy2  
  
import libnum  
  
  
  
p = 11387480584909854985125335848240384226653929942757756384489381242206157197986555243995335158328781970310603060671486688856263776452654268043936036556215243  
  
q = 12972222875218086547425818961477257915105515705982283726851833508079600460542479267972050216838604649742870515200462359007315431848784163790312424462439629  
  
c = 95272795986475189505518980251137003509292621140166383887854853863720692420204142448424074834657149326853553097626486371206617513769930277580823116437975487148956107509247564965652417450550680181691869432067892028368985007229633943149091684419834136214793476910417359537696632874045272326665036717324623992885  
  
dp = 8191957726161111880866028229950166742224147653136894248088678244548815086744810656765529876284622829884409590596114090872889522887052772791407131880103961  
  
dq = 3570695757580148093370242608506191464756425954703930236924583065811730548932270595568088372441809535917032142349986828862994856575730078580414026791444659  
  
def decrypt(dp, dq, p, q, c):  
  
 InvQ = gmpy2.invert(q, p)  
  
 mp = pow(c, dp, p)  
  
 mq = pow(c, dq, q)  
  
 m = (((mp - mq) \* InvQ) % p) \* q + mq  
  
 print(mp - mq)  
  
 print(libnum.n2s(int(m)).decode())  
  
  
  
  
  
decrypt(dp, dq, p, q, c)  
  
# m=pow(c,dp,p)  
  
# m=pow(c,dq,q)  
  
# #前提就是 m<p,m<q  
  
# print(libnum.n2s(m))

### hash



进行MD5依次查询即可



### 一拍即合

根据题意分析进行异或

**m 转换为二进制：** 使用 bin(int(m, 16)) 将十六进制明文转换为二进制，并使用 .zfill() 补齐至 4 位的倍数。

**key 转换为二进制：** 对每个字符进行 ord(c) 转为 ASCII 码，并将其转换为 8 位二进制。

**秘钥补齐：** 如果 key 的二进制长度小于 m 的长度，前面用零补齐。

**秘钥循环扩展：** 如果 key 的长度小于 m，重复 key 直到它的长度与 m 相等。

**按位异或操作：** 对 m 和 key 的二进制按位异或，得到解密后的二进制字符串。

**二进制转字符串：** 每 8 位二进制转换为一个字符，通过 bin\_to\_str 函数实现。

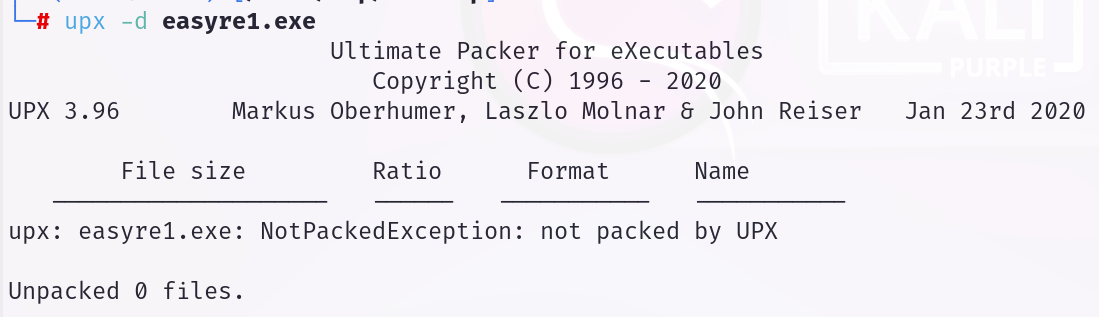


## Reverse

### easyre

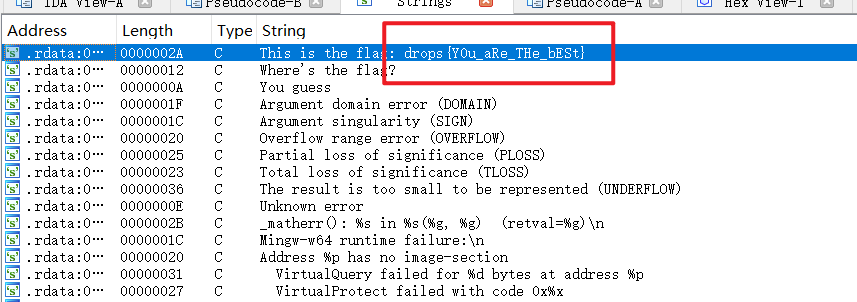
放到IDA中东西不多

拖kali进行脱壳



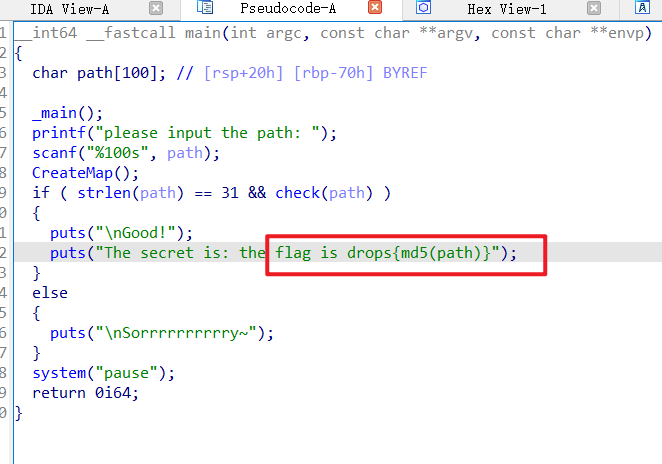
shift+f12

查看字符串



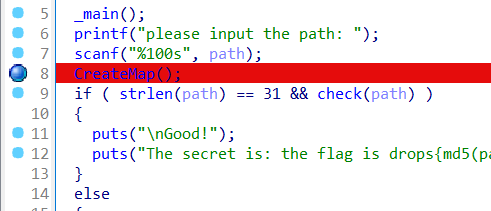
### map

拖IDA分析

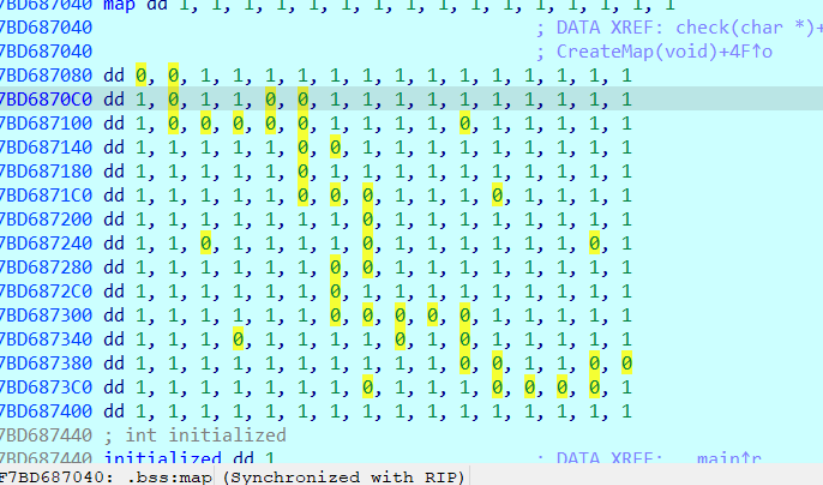


动调跑出地图

设断点，随便输入一个数



调试结束输出地图



手动走地图

dssddddsssddsssassddddssdsdddwd

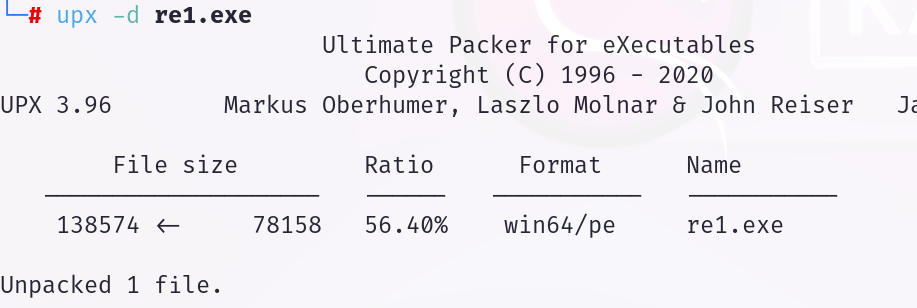
接着md5加密包裹flag即可



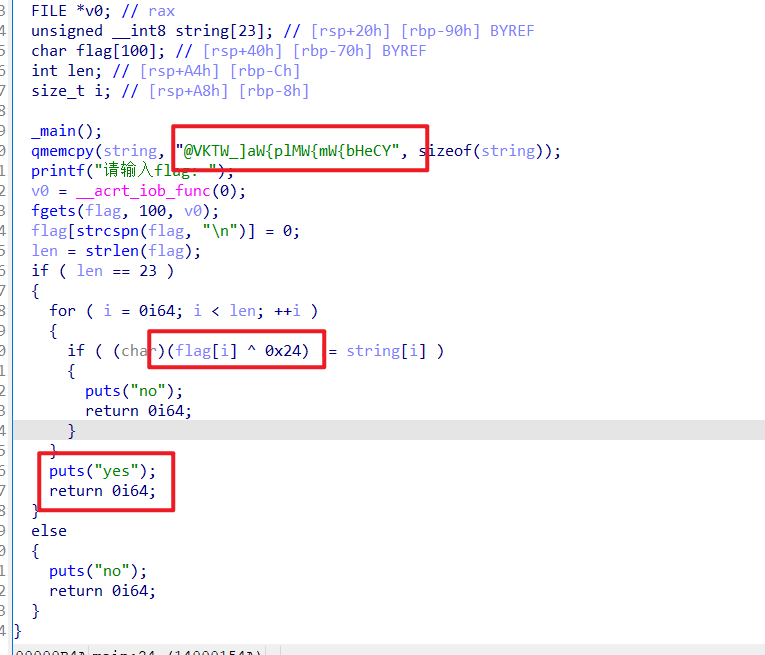
### xor

放到IDA中东西不多

拖kali进行脱壳



再次拖到IDA进行分析



分析加密逻辑

进行异或即可



## Pwn

### This is for you

NC连接后正常操作

