HGAME-Week1-writeup

1.web

1.Hitchhiking_in_the_Galaxy

网页f12发现 /HitchhikerGuide.php , 抓包发现 **302**, 于是在 Linux 中用 curl 命令访问这里,发现 **405** 那么就改用 **post** 请求方式

```
  <center>nginx/1.14.0 (Ubuntu)</center>
  </body>
  </html>

 enc curl-l http://hitchhiker42.0727.site:42420/HitchhikerGuide.php
sh: command not found: curl-l
 enc curl-L http://hitchhiker42.0727.site:42420/HitchhikerGuide.php
sh: command not found: curl-L
 enc curl -l http://hitchhiker42.0727.site:42420/HitchhikerGuide.php
  <html>
  <head><title>405 Method Not Allowed</title></head>
  <body bgcolor="white">
  <center>
      <h1>405 Not Allowed</h1>
      <p>顺风车不是这么搭的</p>
  </center>
  <hr>
  <center>nginx/1.14.0 (Ubuntu)</center>
  </body>
  </html>

 enc curl -X POST http://hitchhiker42.0727.site:42420/HitchhikerGuide.php
  只有使用"无限非概率引擎"(Infinite Improbability Drive)才能访问这里～
```

刚开始看到这句话后没有一点想法,就去看了看发的学习资料,看到 **user-agent** 后意识到应该是要加这个头 (因为引擎) , 内容是那串英文

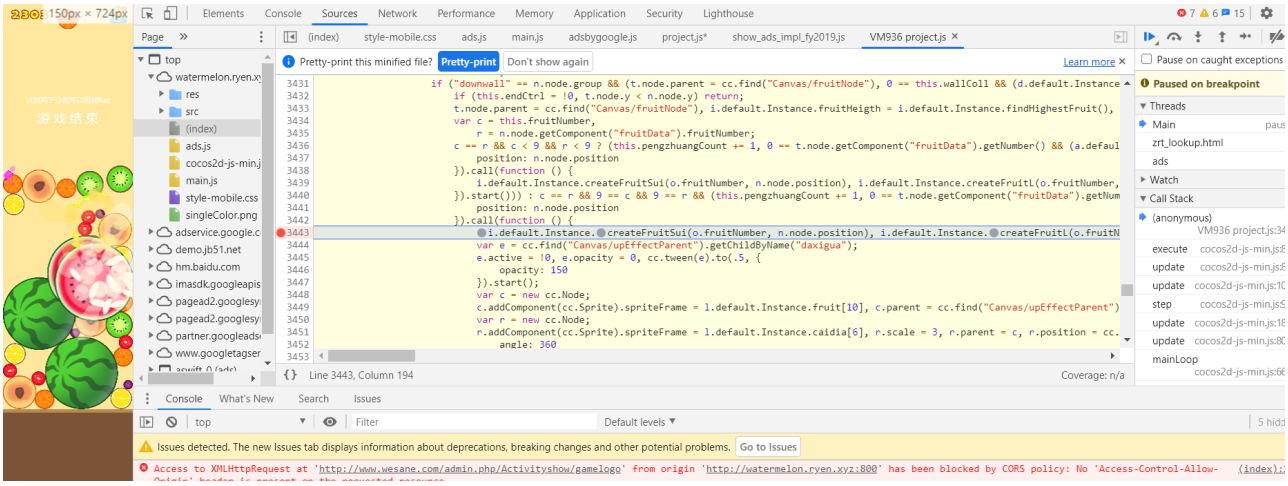
```
Content-Type: text/html; charset=UTF-8

  只有使用"无限非概率引擎"(Infinite Improbability Drive)才能访问这里～
Connection #0 to host hitchhiker42.0727.site left intact
 enc curl -vX POST -H 'User-Agent: Infinite Improbability Drive' -H 'Referer: https://car
al.ink/' http://hitchhiker42.0727.site:42420/HitchhikerGuide.php
  Trying 106.15.250.232...
TCP_NODELAY set
Connected to hitchhiker42.0727.site (106.15.250.232) port 42420 (#0)
POST /HitchhikerGuide.php HTTP/1.1
Host: hitchhiker42.0727.site:42420
Accept: */*
User-Agent: Infinite Improbability Drive
Referer: https://cardinal.ink/

HTTP/1.1 200 OK
Date: Mon, 01 Feb 2021 09:37:22 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 39
Content-Type: text/html; charset=UTF-8

  flag仅能通过本地访问获得
Connection #0 to host hitchhiker42.0727.site left intact
```

仅能通过本地访问 , 那么就再加一个 **x-forwarded-for** 头 , 最后说要从茄子学长的网站过来 , 再加一个 **referer** 头,得到 flag

```
  flag仅能通过本地访问获得
Connection #0 to host hitchhiker42.0727.site left intact
 enc curl -vX POST -H 'User-Agent: Infinite Improbability Drive' -H 'Referer:
ttps://cardinal.ink/' -H 'X-Forwarded-For: 127.0.0.1' http://hitchhiker42.0727.
ite:42420/HitchhikerGuide.php
  Trying 106.15.250.232...
TCP_NODELAY set
Connected to hitchhiker42.0727.site (106.15.250.232) port 42420 (#0)
POST /HitchhikerGuide.php HTTP/1.1
Host: hitchhiker42.0727.site:42420
Accept: */*
User-Agent: Infinite Improbability Drive
Referer: https://cardinal.ink/
X-Forwarded-For: 127.0.0.1

HTTP/1.1 200 OK
Date: Mon, 01 Feb 2021 09:38:37 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 62
Content-Type: text/html; charset=UTF-8

  hgame{s3Cret_0f_HitCHhiking_in_the_GAl@xy_i5_dOnT_p@nic!}
Connection #0 to host hitchhiker42.0727.site left intact
```

2.watermelon

打开网页发现是合成大西瓜, 玩了一遍发现要达到2000分才能得到flag, 于是毫不犹豫地抓包(可惜接受不到分数,是为什么我也不清楚)　抓包不行那么就只能**修改 js 代码**,然而在我试图修改的过程中发现

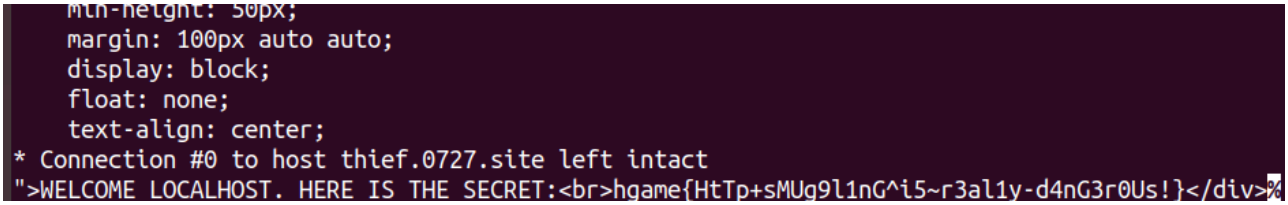


把网页变成这样就能轻松玩到2000分! (到底怎么改js代码我不太清楚,应该是要找到存储分数的变量,然后在 console 里修改.....)

得到 flag

3.宝藏走私者

先拿着学习资料学习了一下,之后点开网页点击 secret data 跳转到另一个界面,提示要是 localhost ,于是加了 host 头



意外的就得到了 flag (不知道怎么得到的, 好像跟学习资料无关?)

4.智商检测鸡

用 firefox 浏览器打开网页发现要做100道定积分才能得到 flag, f12看到网页源代码,里面有个 getflag() 函数

在 console 输入后,弹出一句话(是啥我忘了) , 得知这样不行,于是我就搭配定积分计算器解完了这100题,得到 flag (之后看r4u学长发的学习资料,觉得应该是 python 爬虫一类的,可惜我 python 还不太会,只能手动解题了)

web 总结: web 开始的时候是真的一点都不会, watermelon 和定积分都不是通过 web 知识解的, burp 也是前几天刚装的(安装花了一天...),总之学到了很多

2.reverse

1.apacha

应该是这周 re里最难的（但其实也不是太难），首先用 ida 打开文件并初步分析

```
 9  v8 = __readfsqword(0x28u);
10  v6[0] = 1;
11  v6[1] = 2;
12  v6[2] = 3;
13  v6[3] = 4;
14  sub_11AA();                                   // 打印apacha
15  __printf_chk(1LL, "Please input: ");
16  __isoc99_scanf("%35s", v7);                   // v7输入
17  if ( strlen(v7) != 35 )                       // 长度35
18  {
19    puts("wrong length!");
20    exit(0);
21  }
22  v3 = malloc(0x8CuLL);
23  for ( i = 0LL; i != 35; ++i )
24    v3[i] = v7[i];                              // v3=v7
25  sub_1447(v3, 35, v6);                         // tea加密?
26  if ( sub_1550(v3, 35) )
27    puts("    :)  Flag is your input.");
28  else
29    puts("    :(  Try again.");
30  return 0LL;
31}
```

关键部分在第25和第26行的两个函数

```
v4 = &a1[a2 - 1];                               // 最后一个字符的地址
v5 = *v4;                                       // 最后一个字符
v6 = 0;
do
{
  v6 -= 1640531527;                             // 9e3779b9 tea加密算法特点
  v7 = v6 >> 2;                                 // xxtea
  if ( a2 == 1 )
  {
    v9 = 0;
  }
  else
  {
    v8 = 0LL;
    do
    {
      v5 = a1[v8]
         + ((((v5 >> 5) ^ (4 * a1[v8 + 1])) + ((16 * v5) ^ (a1[v8 + 1] >> 3))) ^ ((*(a3 + 4LL * ((v8 ^ v7) & 3)) ^ v5)
                                                                                 + (a1[v8 + 1] ^ v6)));
      a1[v8++] = v5;
    }
    while ( v8 != (a2 - 2) + 1LL );             // v8!=34
    v9 = a2 - 1;
  }
  result = (16 * v5) ^ (*a1 >> 3);
  v5 = *v4 + (((*(a3 + 4LL * ((v9 ^ v7) & 3)) ^ v5) + (*a1 ^ v6)) ^ (((4 * *a1) ^ (v5 >> 5)) + result));
  *v4 = v5;
}
while ( v6 != -1640531527 * (52 / a2) - 1253254570 );// ?
return result;
}
```

第一个关键函数，刚开始看觉得挺复杂的后来发现了 tea 型加密算法的特征，查资料猜测这应该是 **xxtea** 加密算法

```
unsigned __int64 v2; // rax
int v3; // edx

if ( a2 <= 0 )
  return 1LL;
if ( *a1 != dword_5020 )                        //
                                                //
  return 0LL;
v2 = 4LL;
while ( v2 != 4LL * (a2 - 1) + 4 )
{
  v3 = a1[v2 / 4];                              // 依次取出从a1[1]开始的字符
  v2 += 4LL;
  if ( v3 != *(&unk_501C + v2) )
    return 0LL;
}
return 1LL;
}
```

第二个关键函数，判断输入是否为 flag 。加密后的 flag存在unk_501C , 每四个字节一组

之后就是写脚本解密，于是在网上找了一个现成的脚本（ c语言），改了一下就得到 flag（这里有个地方需要**注意**，因为是**小端序**所以 ida 中加密的数据每四个字节要逆序才是真正的加密数据）

```
#include <stdio.h>
#include <stdint.h>
#define tea_DELTA 0x9e3779b9
#define xxtea_MX (((z>>5^y<<2) + (y>>3^z<<4)) ^ ((sum^y) +
(key[(p&3)^e] ^ z)))
void xxtea(uint32_t* origin, int n, uint32_t const key[4]);
int main()
{
    unsigned int data[] =
    {
0xE74EB323,0xB7A72836,0x59CA6FE2,0x967CC5C1,0xE7802674

,0x3D2D54E6,0x8A9D0356,0x99DCC39C,0x7026D8ED,0x6A33FDAD,

0xF496550A,0x5C9C6F9E,0x1BE5D04C,0x6723AE17,0x5270A5C2,

0xAC42130A,0x84BE67B2,0x705CC779,0x5C513D98,0xFB36DA2D,

0x22179645,0x5CE3529D,0xD189E1FB,0xE85BD489,0x73C8D11F,0x54B5C196,

0xB67CB490,0x2117E4CA,0x9DE3F994,0x2F5AA1AA,0xA7E801FD,0xC30D6EAB,
                        0x1BADDC9C,0x3453B04A,0x92A406F9
    };
```

```
    int i, j;uint32_t* encode = (uint32_t*)data;
    uint32_t const key[4] = {1,2,3,4};
    xxtea(encode, -35, key);
    for (i = 0; i < 35; i++)
        printf("%d,", data[i]);
}
void xxtea(uint32_t* origin, int n, uint32_t const key[4])
{
    uint32_t y, z, sum;
    unsigned p, rounds, e;
    if (n > 1)            /* Coding Part */
    {
        rounds = 6 + 52 / n;
        sum = 0;
        z = origin[n - 1];
        do
        {
            sum += tea_DELTA;
            e = (sum >> 2) & 3;
            for (p = 0; p < n - 1; p++)
            {
                y = origin[p + 1];
                z = origin[p] += xxtea_MX;
            }
            y = origin[0];
            z = origin[n - 1] += xxtea_MX;
        } while (--rounds);
    }
    else if (n < -1)      /* Decoding Part */
    {
        n = -n;
        rounds = 6 + 52 / n;
        sum = rounds * tea_DELTA;
        y = origin[0];
        do
        {
            e = (sum >> 2) & 3;
            for (p = n - 1; p > 0; p--)
            {
                z = origin[p - 1];
                y = origin[p] -= xxtea_MX;
            }
            z = origin[n - 1];
```

```
            y = origin[0] -= xxtea_MX;
            sum -= tea_DELTA;
        } while (--rounds);
    }
}
```

2.helloRe

这题不难，只是有点 c++ 的语法，同样 ida

```
v3 = 0i64;
v14 = 0i64;
v15 = 15i64;
LOBYTE(Block[0]) = 0;
v4 = sub_1400017C0(std::cout, (__int64)"hello, enter your flag please!");
v5 = (__int64 *)std::ostream::operator<<(v4, sub_140001990);
sub_1400017C0(v5, (__int64)"> ");
sub_140001BD0(std::cin, Block);                 // 输入？
sub_1400017C0(std::cout, (__int64)"checking flag ");// 输出字符串的函数？
sub_140001290(200i64);
if ( v14 != 22 )                                // v14=22
LABEL_13:
  sub_140001480();                              // 错误flag
v6 = v15;
v7 = (void **)Block[0];
do
{
  v8 = Block;                                   // v8数组名？
  if ( v6 >= 0x10 )
    v8 = v7;


  if ( (*((_BYTE *)v8 + v3) ^ (unsigned __int8)sub_140001430()) != byte_140003480[v3] )// db 97h, 99h, 9Ch, 91h, 9Eh, 81h, 91h, 9Dh, 9Bh, 2 dup(9Ah)
    goto LABEL_13;                              //  db 0ABh, 81h, 97h, 0AEh, 80h, 83h, 8Fh, 94h, 89h, 99h
                                                // db 97h
  ++v3;
}
while ( v3 < 22 );
v9 = (__int64 *)std::ostream::operator<<(std::cout, sub_140001990);
v10 = sub_1400017C0(v9, (__int64)&unk_140003470);// 输出正确信息
std::ostream::operator<<(v10, sub_140001990);
if ( v6 >= 0x10 )
```

关键在中间的异或操作，写个脚本就能得到结果

```
{
    int i;
    int a[] = { 0x97, 0x99, 0x9C, 0x91, 0x9E, 0x81,
        0x91, 0x9D, 0x9B, 0x9a,0x9a,0x0ab,0x8 (int)158 联机搜索 7,0x0ae,
        0x80,0x83,0x8f,0x94,0x89,0x99,0x97};
    int b = 0x0ff;
    char str[22];
    for (i = 0; i < 22; i++)
    {
        str[i] = a[i] ^ b--;
    }
    printf("%s", str);
    return 0;
```

```
Microsoft Visual Studio 调试控制台
hgame{hello_re_player}诶诶诶诶诶?
C:\Users\18024\Desktop\c\test2\Debug\test2.exe (进程 23012)已退出，代码为 0。
按任意键关闭此窗口. . .
```

3.pypy

这题考的是 python 字节码，刚开始看的时候还不太懂python语法花了点时间去理解，通过 dis 文档来分析每一条语句，大概还原了一下

```
pythonProject – main.py
factor Run Tools VCS Window Help
main
main.py
flag = input("give me your flag:\n")
cipher = list(flag)[6:-1]
length = len(cipher)
res = []
for i in range(length // 2):
    temp = cipher[2 * i]
    cipher[2 * i] = cipher[2 * i + 1]
    cipher[2 * i + 1] = temp

for i in range(length):
    res.append(ord(cipher[i]) ^ i)
```

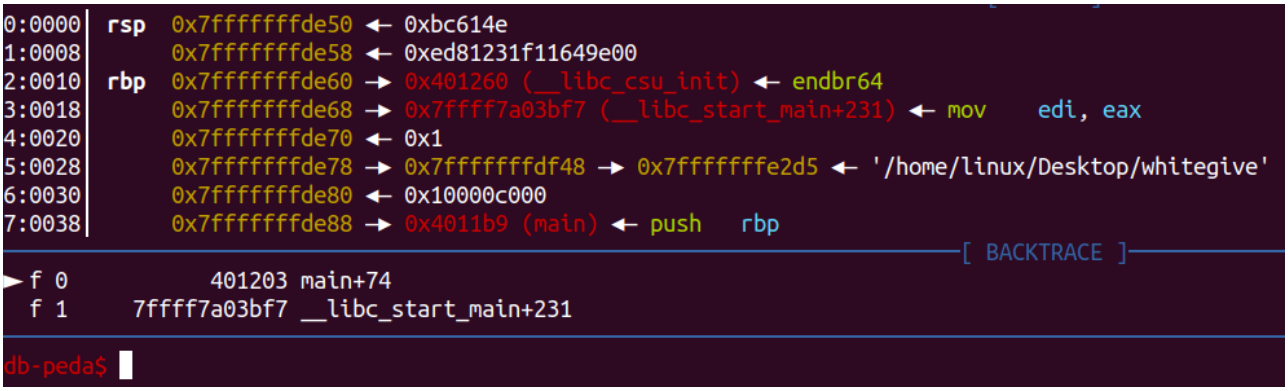
理解之后就是用脚本解题，再加上 hgame{} （刚开始交的时候忘加了）

```
int i,temp;
int a[] = {0x30,0x46,0x66,0x33,0x34,0x6f,0x59,0x21,0x3b,0x41,0x39,
    0x79,0x45,0x20,0x57,0x2b,0x45,0x51,0x4d,0x61,0x58,0x31,0x51,0x57,
    0x66,0x38,0x64,0x3a };
for (i = 0; i < 28; i++)
{
    a[i] = a[i] ^ i;
}
for (i = 0; i < 28/2; i++)
{
    temp = a[i * 2+1];
    a[i * 2+1] = a[i*2];
    a[i * 2] = temp;
}
for (i = 0; i < 28; i++)
{
    printf("%c", a[i]);
}
```

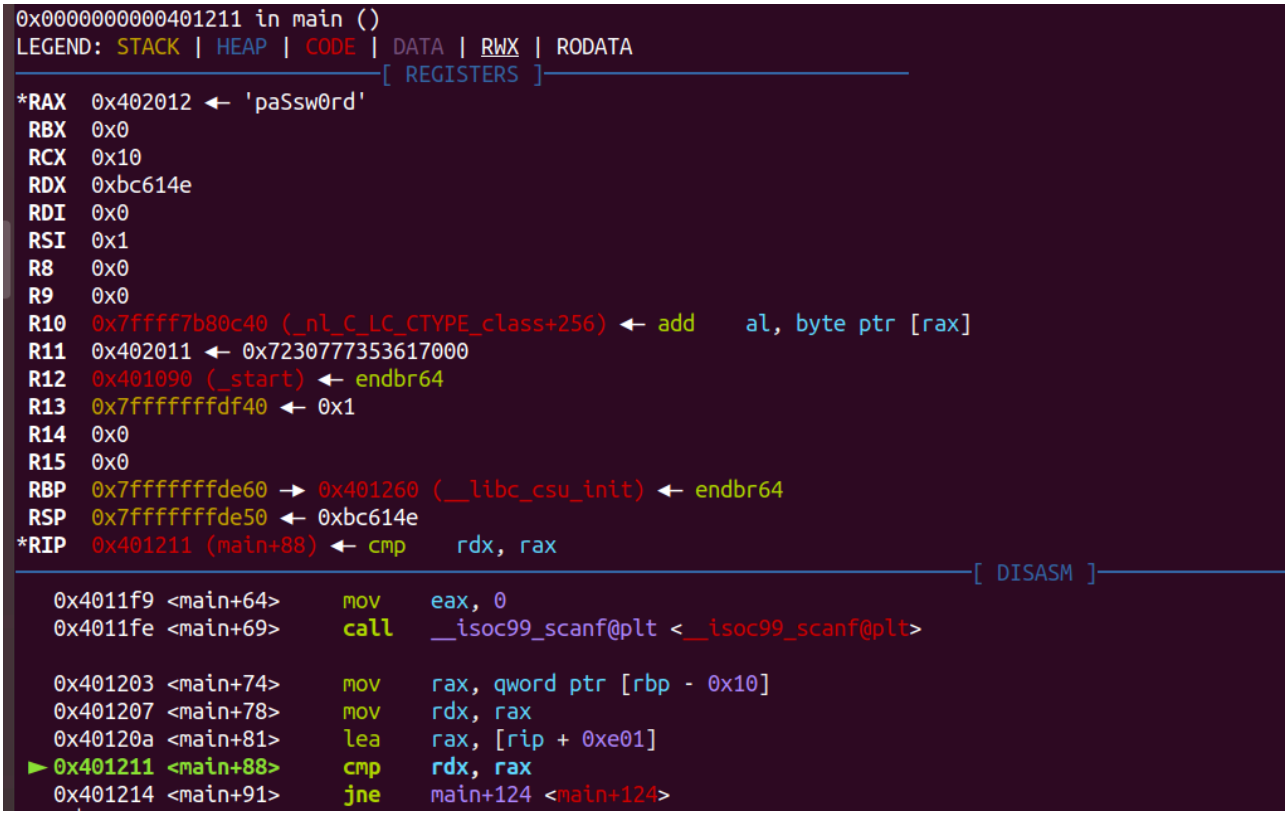
3.pwn

1.whitegive

下载，在 Linux 中用 gdb 调试，得知输入数字在栈中的的存储位置为 0x7fffffffde50



继续调试，发现最后 if 语句比较的是 rax 和 rdx 的值，而此时 rax 存储的是 ‘paSsw0rd’ 字符串的地址，那么只要输入这个地址的十进制值就能使条件成立



```
→ Desktop nc 182.92.108.71 30210
password:4202514
you are right!
ls
bin
dev
flag
lib
lib32
lib64
usr
whitegive
cat flag
hgame{W3lCOme_t0_Hg4m3_2222Z222zO2l}
```

nc连接，cat flag（的确挺白给的，毕竟我都能做）

pwn 总结：我也太菜了只解出白给题 ,看了最后一题但不太会 rop , 总之这周就稍微学习了一下怎么写 pwn 的脚本和一些基础知识以及安装工具

4.crypto

2.对称之美

打开链接下载 python 文件

```python
import random
import string
import itertools
from secret import FLAG

key = ''.join(random.choices(string.ascii_letters + string.digits, k=16))

cipher = bytes([ord(m)^ord(k) for m, k in zip(FLAG, itertools.cycle(key))])

print(cipher)

#cipher=b'_#)\x1d4*\x17"+O-\rA\x16@\x1bu\x19#P.\'\x06>r\x1b,\x06A\x12^\n8\x15>\x04*o\x0c6re%C\x11\x16[\x01!\x19
```

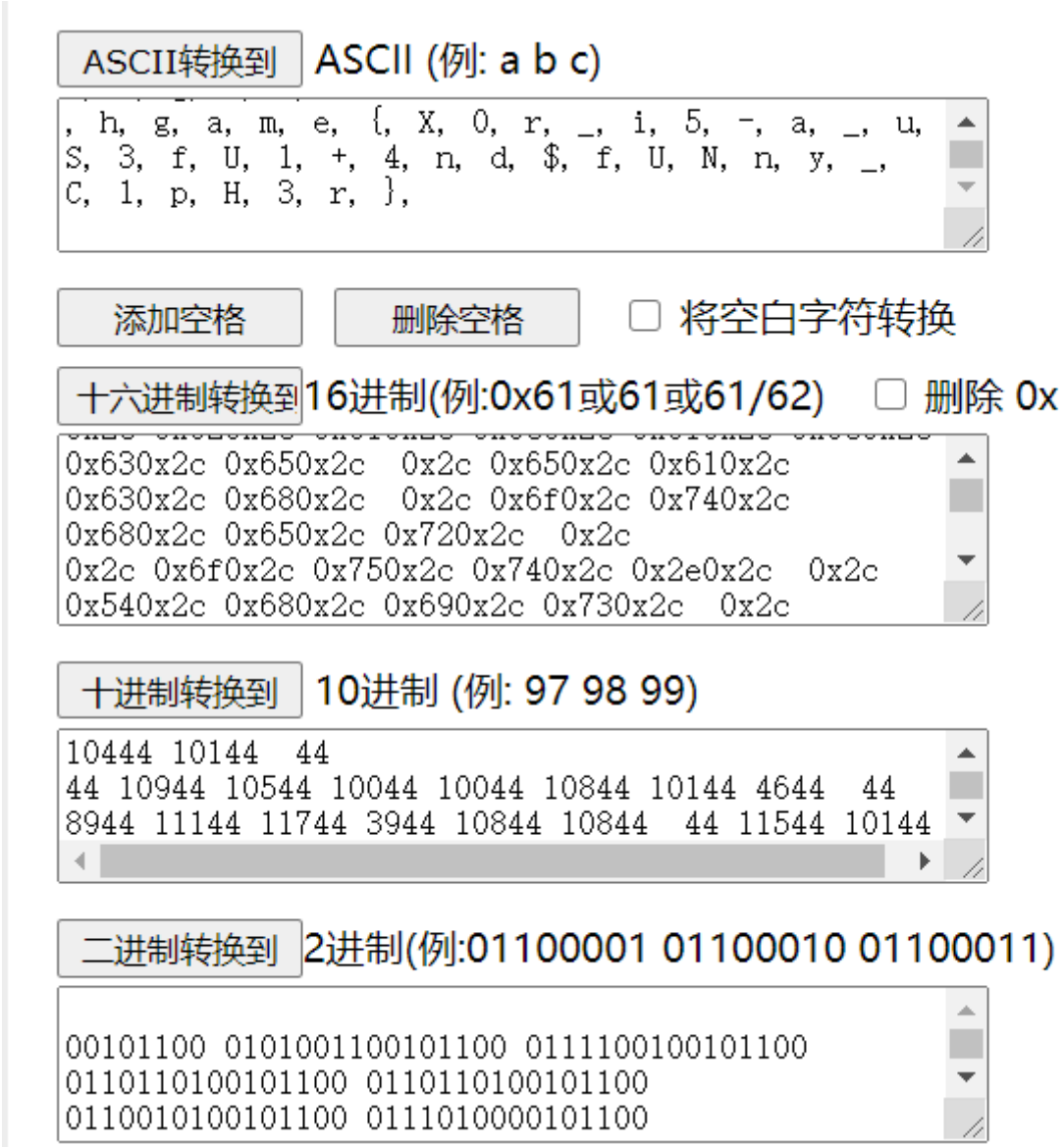
key 是在 ascii 字母和数字里随机选取16个组成，而 cipher 是由明文与循环的 key 异或得到的密文

根据异或特性知道密文与循环的 key 异或就是明文，那么这道题重点就是求出一组 key 。多次下载附件发现 cipher 各不同，那就**利用多个密文**写脚本爆破得到前16位明文

```
table =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
de0 = []
c = []
for i in range(0,130):
    c.append(0)
flag = []
#for i in range(0, 16):
for j in range(0, 62):
    de0.append(xor(cipher0[15], table[j]))
    de0.append(xor(cipher1[15], table[j]))
    de0.append(xor(cipher2[15], table[j]))
    de0.append(xor(cipher3[15], table[j]))
    de0.append(xor(cipher4[15], table[j]))
    de0.append(xor(cipher5[15], table[j]))
    de0.append(xor(cipher6[15], table[j]))
    de0.append(xor(cipher7[15], table[j]))
    de0.append(xor(cipher8[15], table[j]))
    de0.append(xor(cipher9[15], table[j]))
    de0.append(xor(cipher10[15], table[j]))
    de0.append(xor(cipher11[15], table[j]))
    de0.append(xor(cipher12[15], table[j]))
    de0.append(xor(cipher13[15], table[j]))
    de0.append(xor(cipher14[15], table[j]))
    de0.append(xor(cipher15[15], table[j]))
    de0.append(xor(cipher16[15], table[j]))
    de0.append(xor(cipher17[15], table[j]))
    de0.append(xor(cipher18[15], table[j]))
    de0.append(xor(cipher19[15], table[j]))
    de0.append(xor(cipher20[15], table[j]))
    de0.append(xor(cipher21[15], table[j]))
    de0.append(xor(cipher22[15], table[j]))
    de0.append(xor(cipher23[15], table[j]))
    de0.append(xor(cipher24[15], table[j]))
    de0.append(xor(cipher25[15], table[j]))
    de0.append(xor(cipher26[15], table[j]))
    de0.append(xor(cipher27[15], table[j]))
    de0.append(xor(cipher28[15], table[j]))
    de0.append(xor(cipher29[15], table[j]))
    de0.append(xor(cipher30[15], table[j]))
    de0.append(xor(cipher31[15], table[j]))
    de0.append(xor(cipher32[15], table[j]))
    de0.append(xor(cipher33[15], table[j]))
```

```
for j in de0:
    c[j] = c[j] + 1

for j in range(0, 128):
    if c[j] == 34:
        print(j)
```

知道前16位明文，那就可以求得一组 key ，并得到 flag，过程同样是写脚本

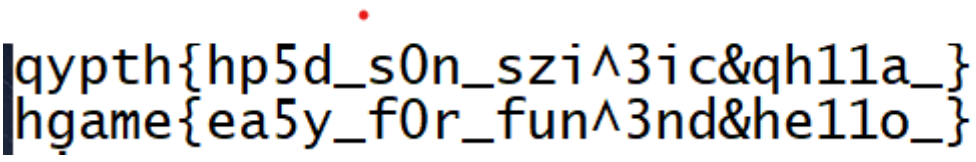


Get flag!

3.Transformer

下载，发现其中一个文件中有flag形式的字符串，还有两个文件分别存储明文和密文，虽然顺序是打乱了的

接下来就采用传统的人工查找明文和密文一一对应的方式得到flag



当然那个 txt 文件最后解密还提示 flag 后面要加上年份2021才正确

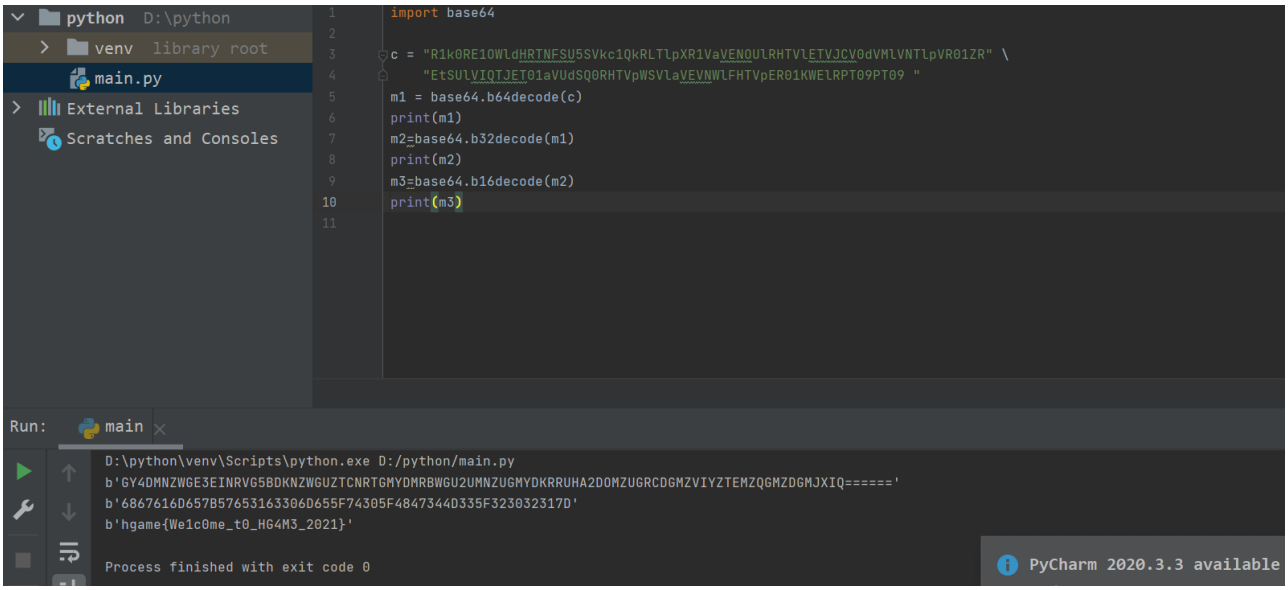
（搜了一下题目发现是人工智能模型，我只有人工没有智能.....）

crypto 总结：没想到最后竟是第一题没解出来，本来以为第二题解不出来反而解出来了。这周稍微学了点怎样写解密脚本，收获挺大

5.misc

1.Base全家福

题目中提到 **base家族**，查资料得知有不只有 base64 ，再根据不同 base 加密字符的特征多次解密



如图所示，得到 flag

2.不起眼压缩包的养成的方法

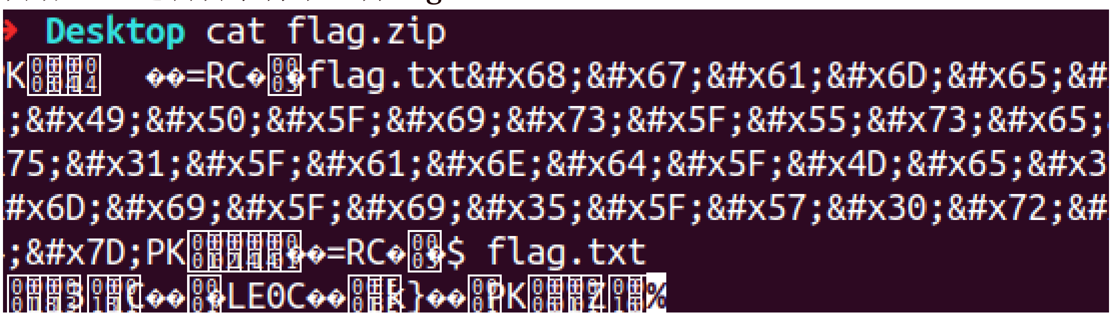
打开得到图片，题目中提到了**压缩包**，猜测能从图片中得到压缩包，于是在 linux 中用 binwalk 检测发现了压缩包，并用 foremost 命令分离出来

分离解压后得到

发现得到里面的文件需要密码，那就用百度识图来获取之前得到的图片的p站id解密。之后点进plain.zip发现又需要密码，并且发现plain.zip里有flag.zip和相同的NO PASSWORD.txt文件，猜测是利用明文攻击解密，用相同的方式压缩已得到的NO PASSWORD.txt。

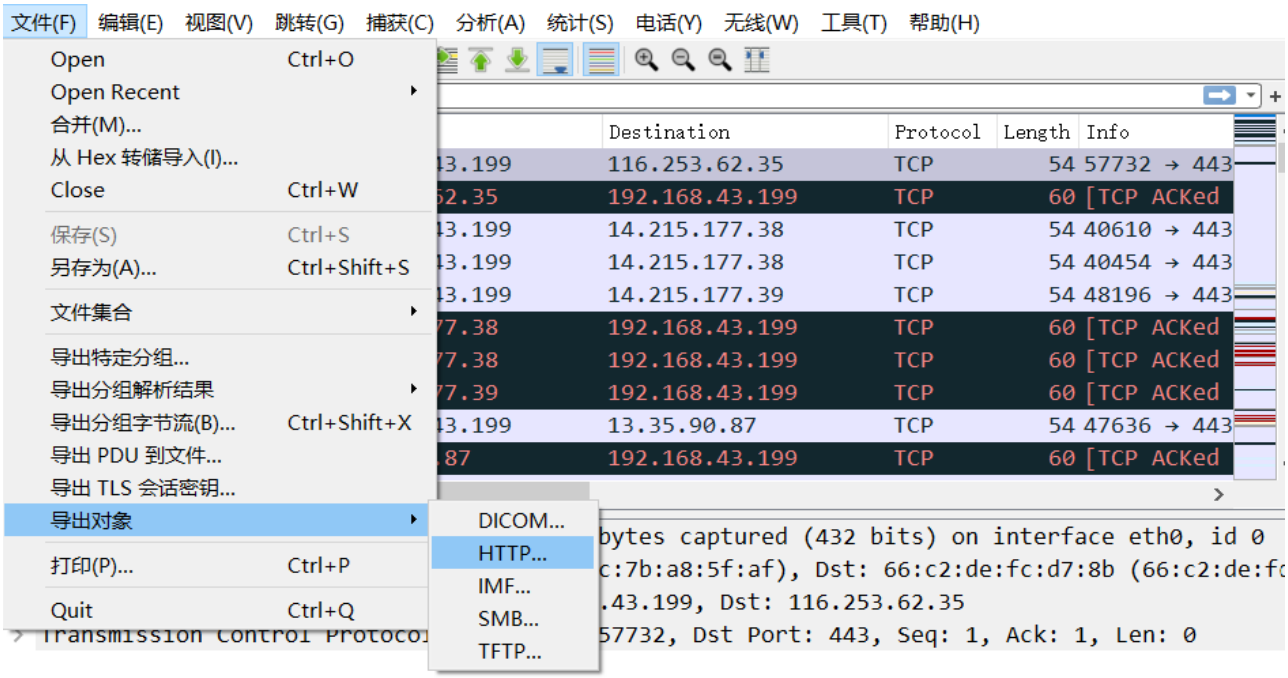
```
Sometimes we don't need to care about password.
Because it's too strong or null. XD
By the way, I only use storage.
```

根据提示**storage**，用**仅存储**的方式压缩（刚开始不知道storage是存储等级所以卡了一会儿，问了学长后才知道） 再用**ARCHPR**进行破解，得到口令和密钥，最后得到flag.zip的内容。16进制转字符串，得flag

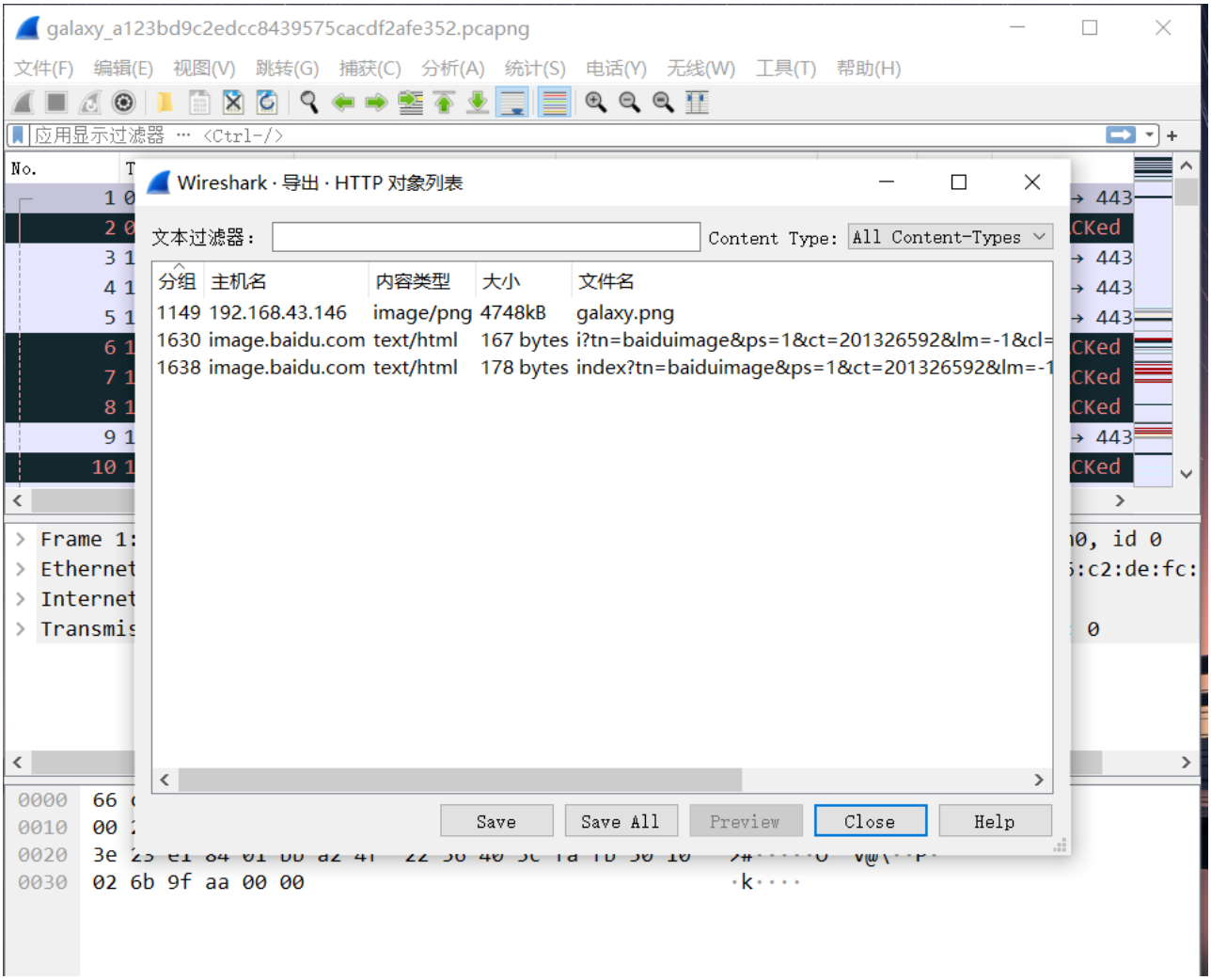


3.Galaxy

打开连接下载，得到以**.pcapng**为后缀的文件，之后用 wireshark 打开



如图所示，导出 http 流



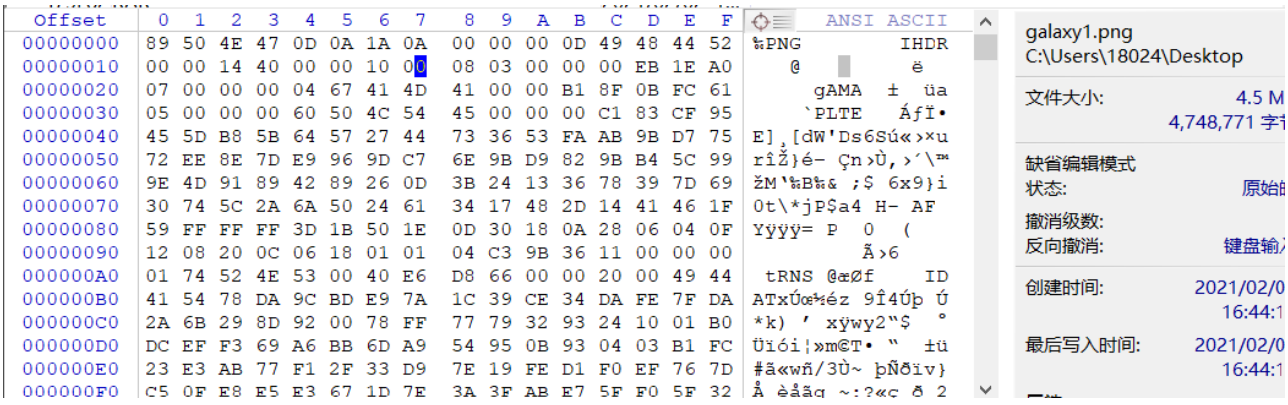
找到了遗失的 galaxy 图片，在 windows 下能正常打开，但之后把图片放到 linux 中，想用 binwalk 分析的时候发现在 Linux 下无法打开图片。

上网查资料得知图片的**高和宽可能被修改**，与原来 crc 值不匹配才报错。于是在网上找了个脚本，得到正确的高宽

```
import struct
import zlib


def hexStr2bytes(s):
    b = b""
    for i in range(0, len(s), 2):
        temp = s[i:i + 2]
        b += struct.pack("B", int(temp, 16))
    return b


str1 = "49484452"
str2 = "0803000000"
bytes1 = hexStr2bytes(str1)
bytes2 = hexStr2bytes(str2)
wid, hei = 5184, 3296

crc32 = "0xEB1EA007"

for w in range(wid, wid + 2000):
    for h in range(hei, hei + 2000):
        width = hex(w)[2:].rjust(8, '0')
```

```
/home/linux/PycharmProjects/pythonProject/venv/bin/python /home/linux/Py
0x1440 0x1000

Process finished with exit code 0
```

然后就用 winhex 修改高和宽并保存



得到 flag！

hgame{Wh4t_A_W0nderfu1_Wa11paper}

4.Word RE:MASTER

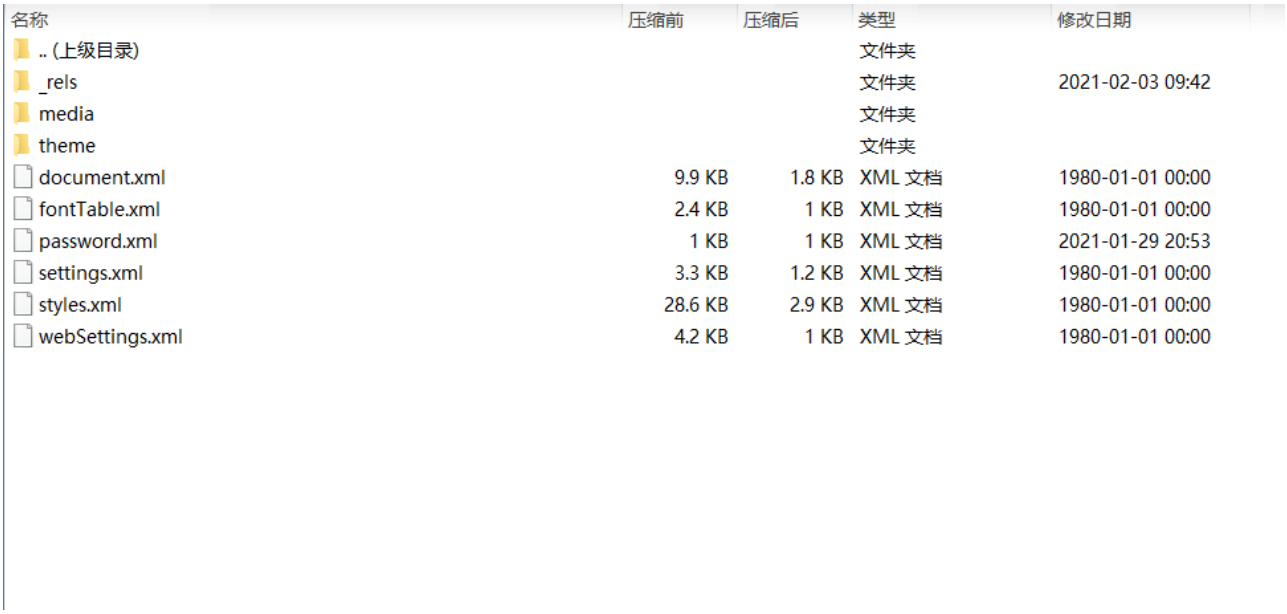
下载得两个文档，发现其中一个文档加密了，那就先打开另一个文档

在 first.docx 文档中发现了 brain 和 fuck 两个单词，猜测有**brainfuck** 加密，但是之后因为不知道密文卡了很久，尝试用 winhex 打开另一个文档得到

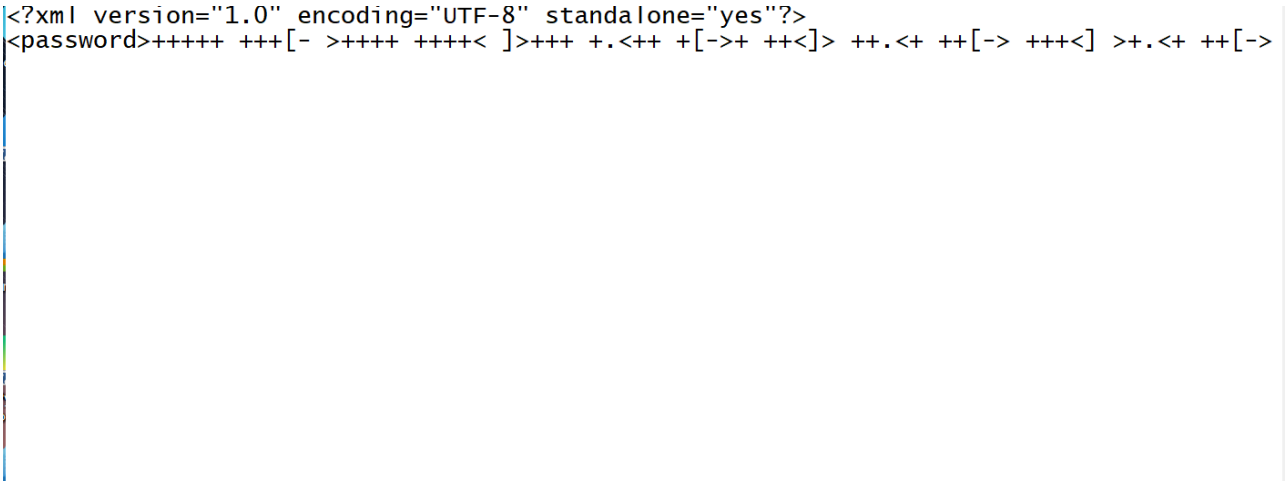
```
0001E7F0  50 61 73 73 77 6F 72 64  20 69 73 20 49 4E 20 74  Password is IN t
0001E800  68 65 20 6F 74 68 65 72  20 64 6F 63 78 2E 00 00  he other docx.
0001E810  20 49 4E 20 74 68 65 20  6F 74 68 65 72 20 64 6F   IN the other do
0001E820  63 78 2E 00 77 6F 72 64  20 49 4E 20 74 68 65 20  cx. word IN the
0001E830  6F 74 68 65 72 20 64 6F  63 78 2E 00 2B 2B 2B 3C  other docx. +++<
0001E840  5D 20 3E 2B 2E 3C 2B 0D  0A 2B 2B 5B 2D 3E 20 2D  ] >+.<+  ++[-> -
0001E850  2D 2D 3C 5D 20 3E 2D 2E  2B 2B 20 2B 2B 2B 2B 2E  --<] >-.++ ++++.
0001E860  20 3C 2B 2B 2B 5B 20 2D  3E 2D 2D 2D 20 3C 5D 3E   <+++[ ->--- <]>
0001E870  2D 2E 20 2B 2B 2B 2E 2B  20 2E 2B 2B 2B 2B 20 2B  -. +++.+ .++++ +
0001E880  2B 2B 2B 2E 20 3C 2B 2B  2B 5B 20 2D 3E 2D 2D 2D  +++. <+++[ ->---
0001E890  0D 0A 3C 5D 3E 2D 2D 20  2D 2D 2D 2D 2E 20 2B 2E    <]>-- ----. +.
0001E8A0  2D 2D 2D 20 2D 2D 2E 2E  2B 20 2E 2B 2B 2B 2B 20  --- --..+ .++++
0001E8B0  2B 2B 2B 2B 2B 20 2E 3C  2B 2B 2B 20 5B 2D 3E 2D  +++++ .<+++ [->-
0001E8C0  2D 20 2D 3C 5D 3E 2D 20  2D 2D 2D 2D 2D 20 2E 3C  - -<]>- ----- .<
0001E8D0  00 00 00 00 1E 00 00 00  06 00 00 00 62 72 61 69              brai
0001E8E0  6E 00 00 00 1E 00 00 00  05 00 00 00 66 75 63 6B  n            fuck
0001E8F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0001E900  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

我以为这就是密文了，于是就开心的拿去解密网站解密（当然这是错的）

问了Akira 学长后发现这密文缺少了前半部分，只能转用别的方法。首先尝试把第一个文档的后缀改为 zip，打开后竟发现



名称	压缩前	压缩后	类型	修改日期
.. (上级目录)			文件夹	
_rels			文件夹	2021-02-03 09:42
media			文件夹	
theme			文件夹	
document.xml	9.9 KB	1.8 KB	XML 文档	1980-01-01 00:00
fontTable.xml	2.4 KB	1 KB	XML 文档	1980-01-01 00:00
password.xml	1 KB	1 KB	XML 文档	2021-01-29 20:53
settings.xml	3.3 KB	1.2 KB	XML 文档	1980-01-01 00:00
styles.xml	28.6 KB	2.9 KB	XML 文档	1980-01-01 00:00
webSettings.xml	4.2 KB	1 KB	XML 文档	1980-01-01 00:00



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<password>+++++ +++[- >++++ ++++< ]>+++ +.<++ +[->+ ++<]> ++.<+ ++[-> +++<] >+.<+ ++[->
```

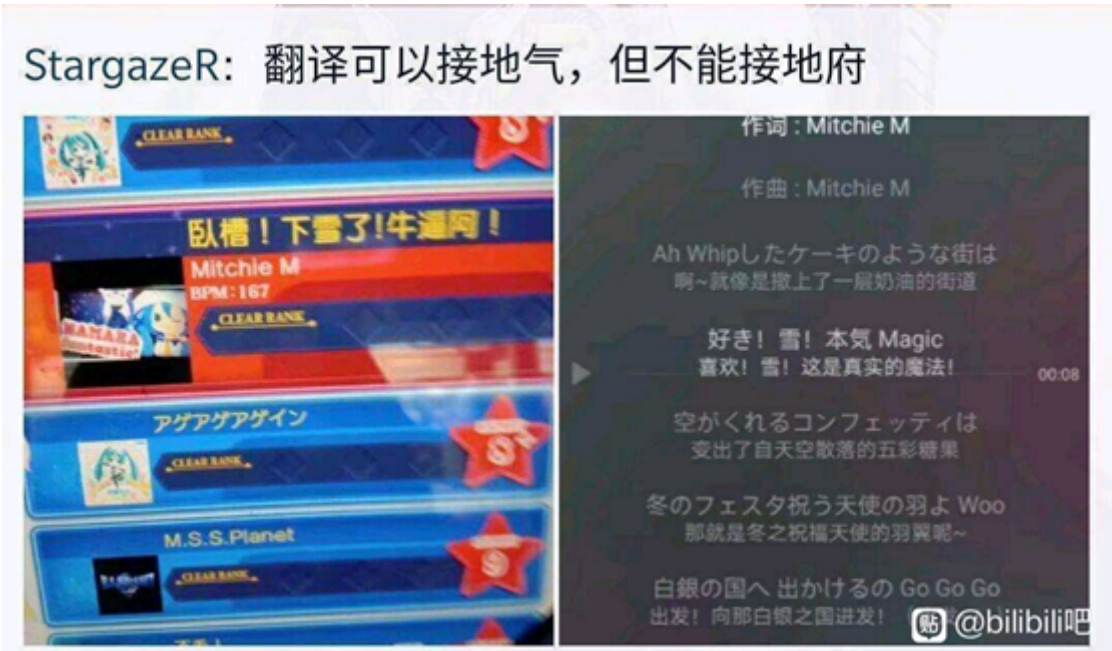
丢到网站解密后得到 DOYOUKNOWHIDDEN?，用这个打开第二个文档

因为这个文档密码是 DOYOUKNOWHIDDEN?，所以立马显示 word 的隐藏字符，得到

→ · → · · → · · · · · · · → · · → · · · · · → · · · · · → · · · → · · · → · · ↵
· → · · · · · · · → · → · · · · → · · · · · · → → · · · · · · · → → · · · · · → · · ↵
· → · · · · · → · · · · · → · · · · → · · · · · · → · · · → → · · → · · → · ↵
· · · · → · → · · · · · → · · · · · · · → · · · · · · · → · · · → · · · · · → → · → · ↵
· · → · · · · · · → · → · → · · · · · → · · → · · · → · · · · · · · → · · · · · · · → · ↵
· · → · · · · · · → · → · · · · → · · → · · · · · · → · · · · · → · · · · → · · · · · · → · · · · · ↵
· · · · → · · · · · → · · · · → → · · → · · · · · · → · → · · · · → · · · · · → · · · · ↵
· · · → · · · · · · · → · · · · · · · → → · · → · · · · · · → · → · · · · → · · → · · · · · · ↵
· · · · · → · · · · → · · · → · · · → · · → · · · · · · → · · · · · → · · · · → · · · · · → · · · · ↵
→ · · → · · · → → · · → · · · · · → · → → · · · · · → · · · · · · ↵
→ · · · · → · · · · → · · · · → · · · → · · · · · · · → · · ↵
↵

一堆**空白字符**，由空格和制表符组成，这里我用箭头和点表示

这之后就卡了很久（大概一天多.....），我甚至尝试用二进制01表示，实在想不出来了于是去问学长，提示我搜一下第二个文档图片中出现最多的字的英文（之前一直盯着图片内容看，以为hint是音游梗，果然是我想太多orz）



接着就搜索**snow**，知道了空白字符的隐写，上工具解密

```
PS C:\Users\18024> D:\SNOW\SNOW.EXE -C "C:\Users\18024\Desktop\新建文本文档 (2).txt"
hgame{Challen9e_Whit3_P4ND0R4_P4R4D0XXX}
```

misc总结：拿到题目后完全没思路，所以在这一周中查了很多资料，觉得自己就是个菜鸡，好在最后全部解了出来，也学到了很多东西