Misc

1. SignIn

   把图片拉长看出来

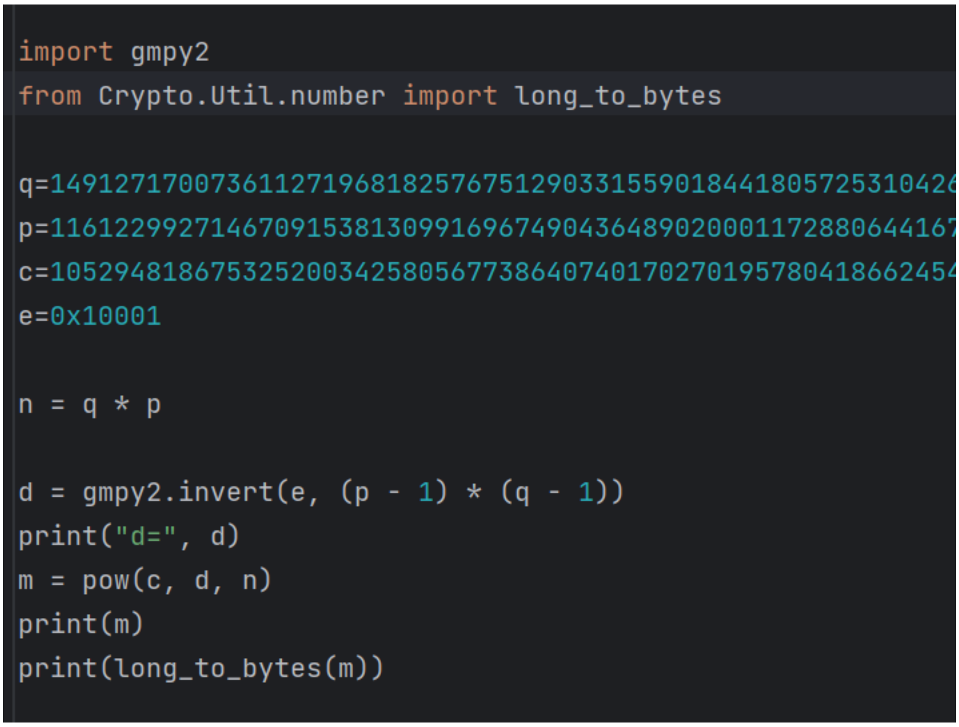
2. 签到

   关注“凌武科技”微信公众号，发送“HGAME2024”获得 Flag!

Crypto

1. ezRSA

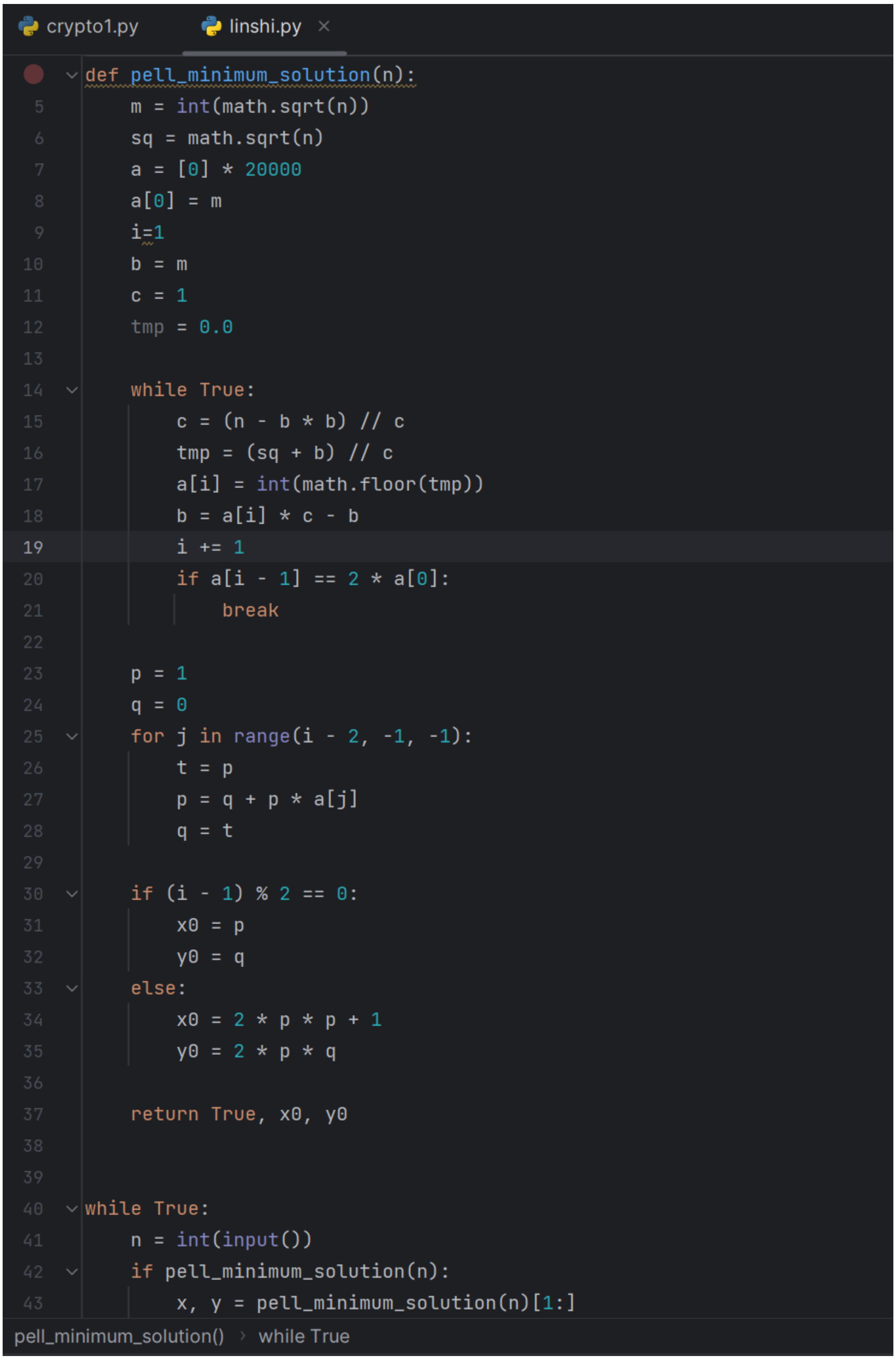
   根据费马小定理，leak1 和 leak2 就是 p 和 q



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

q=149127170073611271968182576751290331559018441805725310426
p=116122992714670915381309916967490436489020001172880644167
c=105294818675325200342580567738640740170270195780418662454
e=0x10001

n = q * p

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

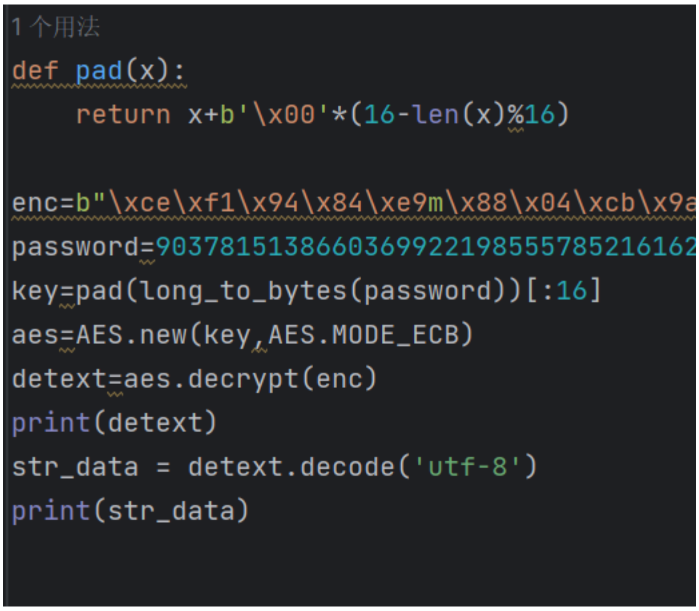
2. ezMath

   大概就是先这样（连分数法解佩尔方程特解）



```python
def pell_minimum_solution(n):
    m = int(math.sqrt(n))
    sq = math.sqrt(n)
    a = [0] * 20000
    a[0] = m
    i=1
    b = m
    c = 1
    tmp = 0.0

    while True:
        c = (n - b * b) // c
        tmp = (sq + b) // c
        a[i] = int(math.floor(tmp))
        b = a[i] * c - b
        i += 1
        if a[i - 1] == 2 * a[0]:
            break

    p = 1
    q = 0
    for j in range(i - 2, -1, -1):
        t = p
        p = q + p * a[j]
        q = t

    if (i - 1) % 2 == 0:
        x0 = p
        y0 = q
    else:
        x0 = 2 * p * p + 1
        y0 = 2 * p * q

    return True, x0, y0


while True:
    n = int(input())
    if pell_minimum_solution(n):
        x, y = pell_minimum_solution(n)[1:]
```

再那样（解密）



```
1 个用法
def pad(x):
    return x+b'\x00'*(16-len(x)%16)

enc=b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9a
password=9037815138660369922198555785216162
key=pad(long_to_bytes(password))[:16]
aes=AES.new(key,AES.MODE_ECB)
detext=aes.decrypt(enc)
print(detext)
str_data = detext.decode('utf-8')
print(str_data)
```

Pwn

1. EzSignIn

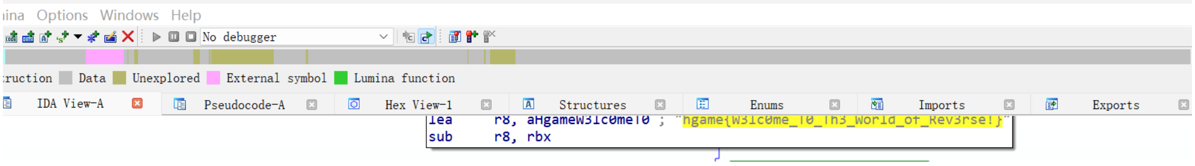
   终端输入 nc 地址 端口

2. Ezshellcode

Reverse

1. ezIDA

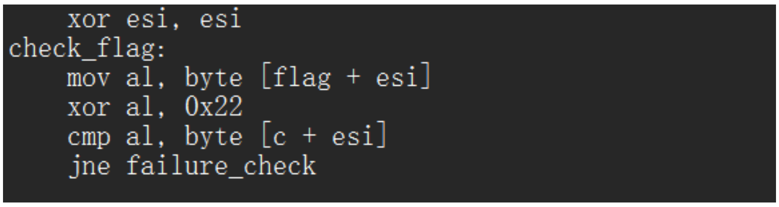
   打开 ida



```
lea     r8, aHgameW31c0meT0 ; "hgame{W31c0me_T0_Th3_World_of_Rev3rse!}"
sub     r8, rbx
```

2. ezASM

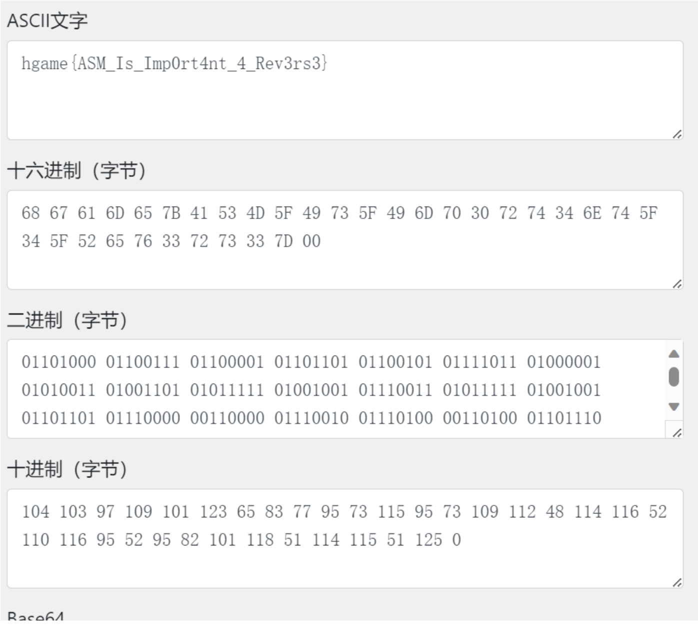
   这里看出 flag 长为 33 个字节，然后与 34 异或



```
    xor esi, esi
check_flag:
    mov al, byte [flag + esi]
    xor al, 0x22
    cmp al, byte [c + esi]
    jne failure_check
```

   再和 34 异或一次就是原来的数了

   （这里我是原始人，一个一个让电脑算的）



ASCII文字

```
hgame{ASM_Is_Imp0rt4nt_4_Rev3rs3}
```

十六进制（字节）

```
68 67 61 6D 65 7B 41 53 4D 5F 49 73 5F 49 6D 70 30 72 74 34 6E 74 5F
34 5F 52 65 76 33 72 73 33 7D 00
```

二进制（字节）

```
01101000 01100111 01100001 01101101 01100101 01111011 01000001
01010011 01001101 01011111 01001001 01110011 01011111 01001001
01101101 01110000 00110000 01110010 01110100 00110100 01101110
```

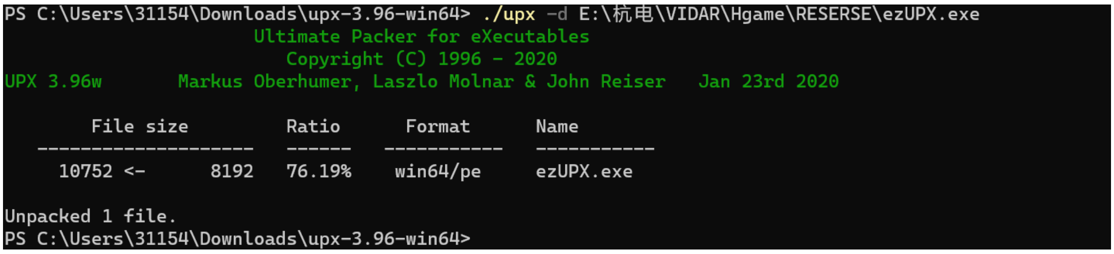
十进制（字节）

```
104 103 97 109 101 123 65 83 77 95 73 115 95 73 109 112 48 114 116 52
110 116 95 52 95 82 101 118 51 114 115 51 125 0
```

Base64

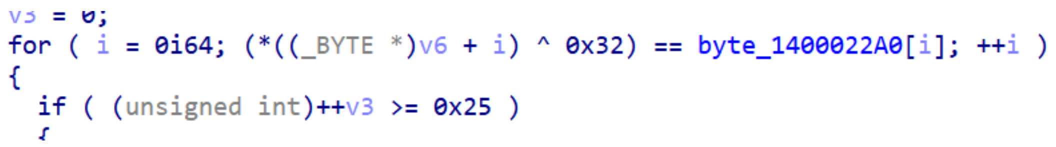
3. ezUPX

先这样



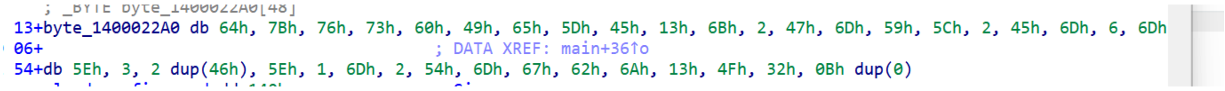
```
PS C:\Users\31154\Downloads\upx-3.96-win64> ./upx -d E:\杭电\VIDAR\Hgame\RESERSE\ezUPX.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2020
UPX 3.96w       Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     10752 <-      8192   76.19%    win64/pe     ezUPX.exe

Unpacked 1 file.
PS C:\Users\31154\Downloads\upx-3.96-win64>
```

F5 看到异或了 0x32



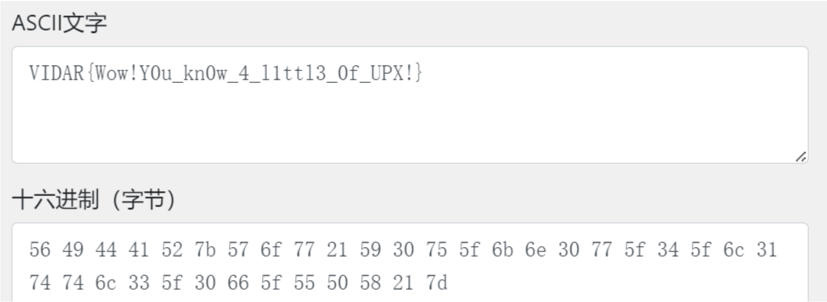
```
v3 = 0;
for ( i = 0i64; (*((_BYTE *)v6 + i) ^ 0x32) == byte_1400022A0[i]; ++i )
{
  if ( (unsigned int)++v3 >= 0x25 )
  {
```

点进字节串



```
    ; _BYTE byte_1400022A0[48]
13+byte_1400022A0 db 64h, 7Bh, 76h, 73h, 60h, 49h, 65h, 5Dh, 45h, 13h, 6Bh, 2, 47h, 6Dh, 59h, 5Ch, 2, 45h, 6Dh, 6, 6Dh
06+                                         ; DATA XREF: main+36↑o
54+db 5Eh, 3, 2 dup(46h), 5Eh, 1, 6Dh, 2, 54h, 6Dh, 67h, 62h, 6Ah, 13h, 4Fh, 32h, 0Bh dup(0)
```

原始人再次手算



ASCII文字

```
VIDAR{Wow!Y0u_kn0w_4_l1ttl3_0f_UPX!}
```

十六进制（字节）

```
56 49 44 41 52 7b 57 6f 77 21 59 30 75 5f 6b 6e 30 77 5f 34 5f 6c 31
74 74 6c 33 5f 30 66 5f 55 50 58 21 7d
```