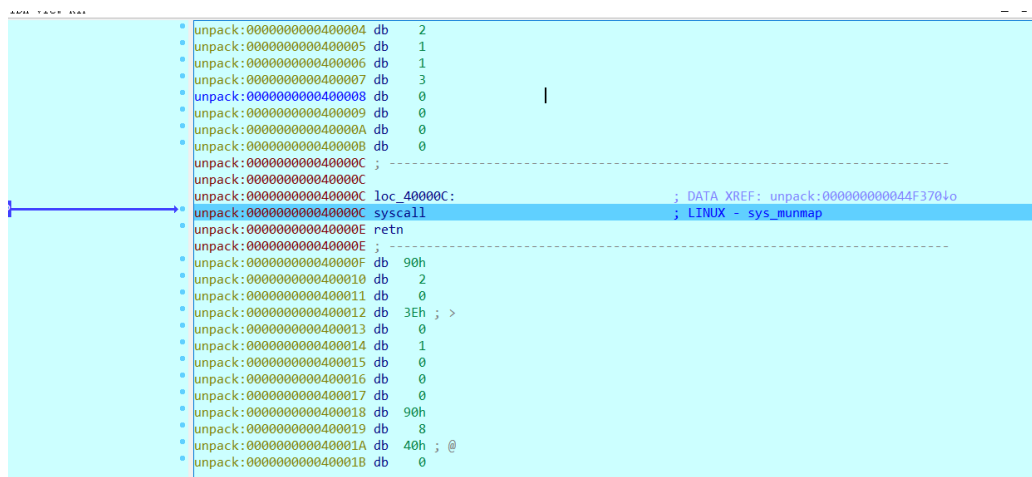


HGAME Week2 WriteUp

Unpack

1. 用 exeinfope 可以看出是加了 UPX 的壳的
2. 然后百度到手解 UPX 壳的方式，丢到 IDA 里调试



3. 在这之后用 dump 脚本即可得到脱壳后的代码
4. 再用 IDA 打开新的代码 F12 后根据 flag 查到相应代码

```
3  const char *v0; // rdi
4  __int64 v1; // rdx
5  __int64 result; // rax
6  __int64 v3; // rcx
7  unsigned __int64 v4; // rt1
8  signed int v5; // [rsp+8h] [rbp-48h]
9  signed int i; // [rsp+Ch] [rbp-44h]
10 char v7[56]; // [rsp+10h] [rbp-40h]
11 unsigned __int64 v8; // [rsp+48h] [rbp-8h]
12
13 v8 = __readfsqword(0x28u);
14 sub_40F570((unsigned __int64)"%42s");
15 v5 = 0;
16 for ( i = 0; i <= 41; ++i ) char[5]
17 {
18     if ( i + v7[i] != (unsigned __int8)byte_6CA0A0[i] )
19         v5 = 1;
20 }
21 if ( v5 == 1 )
22 {
23     v0 = "Wrong input";
24     sub_40FE40("Wrong input", v7);
25 }
26 else
27 {
28     v0 = "Congratulations! Flag is your input";
29     sub_40FE40("Congratulations! Flag is your input", v7);
30 }
31 result = 0LL;
32 v4 = __readfsqword(0x28u);
33 v3 = v4 ^ v8;
34 if ( v4 != v8 )
35     sub_443040(v0, v7, v1, v3);
36 return result;
37 }
```

5. 很容易看出是按位加密

6. 写个脚本就能得出 flag

babyPy

1. 直接观察可知

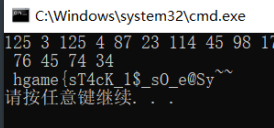
4 是 对字符串的倒序排列 $000 = 000[::-1]$

5 是 元组和列表的转换

6--8 对字符串加密 加密的方法是与前一位进行异或处理

2. 根据脚本可以直接得出 flag

```
{
char * num = "7d037d045717722d62114e6a5b044f2c184c3f44214c2d4a22";
int nu[30] = {0};
for(int i=0;i<strlen(num)/2;i++)
{
    int a,b;
    if( 'a' <= num[i*2] && num[i*2] <= 'f')
    {
        a = num[i*2] - 'a' + 10;
    }
    else
    {
        a = num[i*2] - '0';
    }
    if( 'a' <= num[i*2+1] && num[i*2+1] <= 'f')
    {
        b = num[i*2+1] - 'a' + 10;
    }
    else
    {
        b = num[i*2+1] - '0';
    }
    nu[i] = a * 16 + b;
    printf("%d ",nu[i]);
}
printf("\n");
for(int i=24;i>0;i--)
{
    nu[i] = nu[i] ^ nu[i-1];
}
for(int i=25;i>0;i--)
{
    printf("%c",nu[i]);
}
}
```



```
C:\Windows\system32\cmd.exe
125 3 125 4 87 23 114 45 98 17
76 45 74 34
hgame{sT4cK_1$_s0_e@Sy~~
请按任意键继续. . .
```