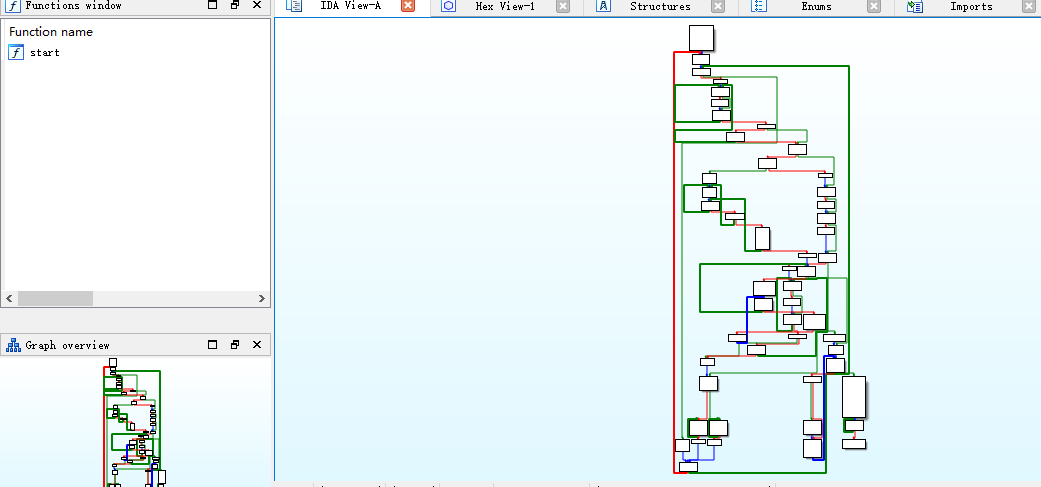
## 题目分析

我们把程序拖入IDA会发现



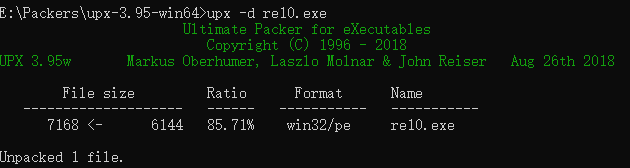
可以判断出程序是加了壳的。

用Exeinfo查看加的壳是UPX壳。

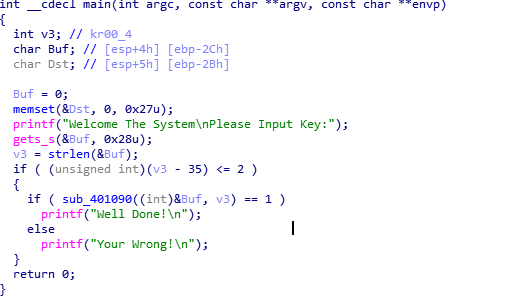


关于脱壳，我们去下UPX\_\_v3.95(<https://down.52pojie.cn/Tools/Packers/>)。

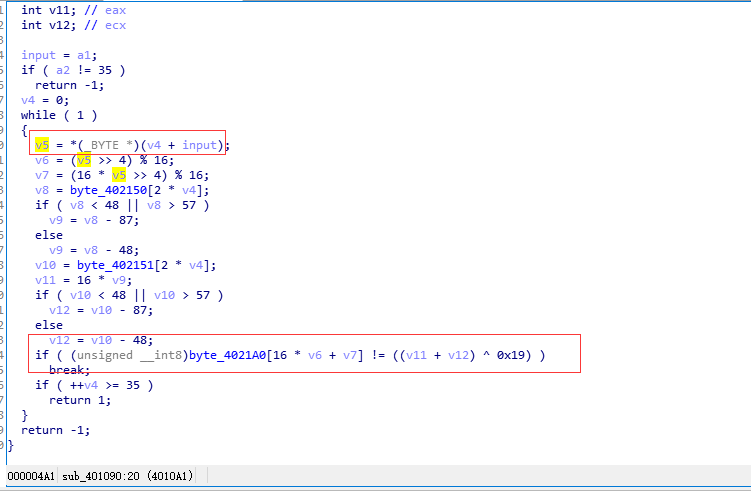
然后进行脱壳。



然后再拖进IDA可以发现



然后可以发现处理输入的函数是在sub\_401090里面，点击进入查看



可以发现函数主要是对输入的每个字符进行处理，然后最后再进行比较。

然后对输入进行的运算和最后比较其实就是if( data[input[i]] == arr[i] ^ 0x19 )，如果看不出来的话也可以进行爆破。

其中data数组是byte\_4021处存放的数组，arr数组是hex字符串“2a49f69c38395cde96d6de96d6f4e025484954d6195448def6e2dad67786e21d5adae6”转化为bytes的数组。

## 脚本

data = [99, 124, 119, 123, 242, 107, 111, 197, 48, 1, 103, 43, 254, 215, 171, 118, 202, 130, 201, 125, 250, 89, 71, 240, 173, 212, 162, 175, 156, 164, 114, 192, 183, 253, 147, 38, 54, 63, 247, 204, 52, 165, 229, 241, 113, 216, 49, 21, 4, 199, 35, 195, 24, 150, 5, 154, 7, 18, 128, 226, 235, 39, 178, 117, 9, 131, 44, 26, 27, 110, 90, 160, 82, 59, 214, 179, 41, 227, 47, 132, 83, 209, 0, 237, 32, 252, 177, 91, 106, 203, 190, 57, 74, 76, 88, 207, 208, 239, 170, 251, 67, 77, 51, 133, 69, 249, 2, 127, 80, 60, 159, 168, 81, 163, 64, 143, 146, 157, 56, 245, 188, 182, 218, 33, 16, 255, 243, 210, 205, 12, 19, 236, 95, 151, 68, 23, 196, 167, 126, 61, 100, 93, 25, 115, 96, 129, 79, 220, 34, 42, 144, 136, 70, 238, 184, 20, 222, 94, 11, 219, 224, 50, 58, 10, 73, 6, 36, 92, 194, 211, 172, 98, 145, 149, 228, 121, 231, 200, 55, 109, 141, 213, 78, 169, 108, 86, 244, 234, 101, 122, 174, 8, 186, 120, 37, 46, 28, 166, 180, 198, 232, 221, 116, 31, 75, 189, 139, 138, 112, 62, 181, 102, 72, 3, 246, 14, 97, 53, 87, 185, 134, 193, 29, 158, 225, 248, 152, 17, 105, 217, 142, 148, 155, 30, 135, 233, 206, 85, 40, 223, 140, 161, 137, 13, 191, 230, 66, 104, 65, 153, 45, 15, 176, 84, 187, 22]  
arr = "2a49f69c38395cde96d6de96d6f4e025484954d6195448def6e2dad67786e21d5adae6".decode('hex')  
flag = []  
for i in range(len(arr)):  
 value = ord(arr[i]) ^ 0x19  
 flag.append(data.index(value))  
print ''.join(map(chr,flag))

**flag{Th1s\_1s\_Simple\_Rep1ac3\_Enc0d3}**