XCTF EasyRE

原创

酸酸菜鱼 于 2020-07-31 18:15:07 发布 298 收藏
分类专栏： CTF

本文链接：https://blog.csdn.net/lhk124/article/details/107718508

直接看ida和代码

```
sub_401020((int)&unk_402150);
v9 = 0;
v10 = 0;
v7 = 0i64;
v8 = 0i64;
sub_401050((int)&unk_402158, &v7);              // %s
v0 = strlen((const char *)&v7);
if ( v0 >= 0x10 && v0 == 0x18 )                 // v0大于16且等于24，v0是v7的长度
{
  v1 = 0;
  v2 = (char *)&v8 + 7;                         // 做循环，该地方的每个位置的值为v3，v3 = v2--
  do
  {
    v3 = *v2--;
    input[v1++] = v3;
  }
  while ( v1 < 0x18 );                          // 直至循环结束
  v4 = 0;
  do
  {
    input[v4] = (input[v4] + 1) ^ 6;            // 输入+1后异或6，循环
    ++v4;
  }
  while ( v4 < 0x18 );                          // 直至24
  v5 = strcmp(input, (const char *)&unk_402124);// 人间迷惑行为：这里不是那个flag，是下边那段
  if ( v5 )
    v5 = -(v5 < 0) | 1;
  if ( !v5 )
  {
    sub_401020((int)"right\n");
    system("pause");
  }
}
return 0;
```

```
str_list = [0x78,0x49,0x72,0x43,0x6A,0x7E,0x3C,0x72,0x7C,0x32,0x74,0x57,0x73,0x76,0x33,
 0x50,0x74,0x49,0x7F,0x7A,0x6E,0x64,0x6B,0x61]
flag_list = []
# print(len(str_list))

"""
正向：1.判断长度是否为24
     2.对输入的每个字符串从最后一位开始往前进行+1,^0x6
     3.与上边的字符串比较
逆向：用上边的字符串 ^0x6 后 -1 得出正确的原始输入
"""
for i in range(len(str_list)):
    flag = (str_list[-i-1] ^ 0x6) - 1  # -i-1 调整的是每个字符的位置 （i初始为0，但倒序是从-1开始才对的）
    flag_list.append(chr(flag))

print("".join(flag_list))

# flag{xNqU4otPq3ys9wkDsN}
```

点

- 放个假的flag，在靠近真实内容的地方，混淆
- F5查看时眼花了，因为他用了[v4]又+1，容易误认为是取下一位这样的循环，实际是+1
- 可爆破，没爆破的必要。

```
00A91100  |> /8A06          /mov al,byte ptr ds:[esi]                ;  指向最后一个
00A91102  |. |8D76 FF       |lea esi,dword ptr ds:[esi-0x1]          ;  往前
00A91105  |. |8882 6C33A900 |mov byte ptr ds:[edx+0xA9336C],al
00A9110B  |. |42            |inc edx
00A9110C  |. |3BD1          |cmp edx,ecx
00A9110E  |.^\7C F0         \jl short 210f1e18.00A91100

从最后一位开始取的汇编代码
```

整一下两条命令的含义

00A910F6 |. 66   datasize:
00A910F7 |. 66:0f1f8400 0>nop word ptr ds:[eax+eax]