

原创

北风~ 于 2020-04-29 22:23:57 发布 1137 收藏 1

分类专栏: [逆向与保护](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_45055269/article/details/105850121

版权



[逆向与保护](#) 专栏收录该内容

65 篇文章 4 订阅

订阅专栏

工具

IDA动调

思路展开

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    char s; // [rsp+0h] [rbp-20h]
    int v5; // [rsp+18h] [rbp-8h]
    int i; // [rsp+1Ch] [rbp-4h]

    for ( i = 0; i <= 181; ++i )
    {
        envp = (const char *)*((unsigned __int8 *)judge + i) ^ 0xCu;
        *((_BYTE *)judge + i) ^= 0xCu;
    }
    printf("Please input flag:", argv, envp);
    __isoc99_scanf("%20s", &s);
    v5 = strlen(&s);
    if ( v5 == 14 && (unsigned int)judge(&s) ) #输入长度14
        puts("Right!");
    else
        puts("Wrong!");
    return 0;
}
```

judge函数F5没用，猜测judge函数动态生成，下断点动调。

```
0B04 mov     [rbp-28h], rdi
0B08 mov     byte ptr [rbp-20h], 66h
0B08 judge endp ; sp-analysis failed
0B08
0B0C mov     byte ptr [rbp-1Fh], 6Dh
0B10 mov     byte ptr [rbp-1Eh], 63h
0B14 mov     byte ptr [rbp-1Dh], 64h
0B18 mov     byte ptr [rbp-1Ch], 7Fh
0B1C mov     byte ptr [rbp-1Bh], 6Bh
0B20 mov     byte ptr [rbp-1Ah], 37h
0B24 mov     byte ptr [rbp-19h], 64h
```

https://blog.csdn.net/weixin_45055269

00000000000000000000000000000000: .data:00000000000000000000000000000000 (Synchronized with RIP)

发现14个赋值语句，猜测与flag有关。接着跟进去找到关键算法‘异或’（下图）

```
• .data:00000000000000000000000000000000B4F mov     rax, [rbp-28h]
• .data:00000000000000000000000000000000B53 add     rax, rdx
• .data:00000000000000000000000000000000B56 mov     edx, [rbp-4]
• .data:00000000000000000000000000000000B59 movsxd  rcx, edx
IP .data:00000000000000000000000000000000B5C mov     rdx, [rbp-28h]
• .data:00000000000000000000000000000000B60 add     rdx, rcx
• .data:00000000000000000000000000000000B63 movzx   edx, byte ptr [rdx]
• .data:00000000000000000000000000000000B66 mov     ecx, [rbp-4]
• .data:00000000000000000000000000000000B69 xor     edx, ecx
• .data:00000000000000000000000000000000B6B mov     [rax], dl
• .data:00000000000000000000000000000000B6D add     dword ptr [rbp-4], 1
```

00000B66 00000000000000000000000000000000: .data:00000000000000000000000000000000 (Synchronized with RIP)

Hex View-1 https://blog.csdn.net/weixin_45055269

查看ecx的值发现是0x0到0xe，正好14个，与前面赋值语句异或就是flag。

```
cipher=[0x66,0x6d,0x63,0x64,0x7f,0x6b,0x37,0x64,0x3b,0x56,0x60,0x3b,0x6e,0x70]
flag=""
for i in range(0x0,0xe):
    flag+=chr(cipher[i]^i)
print(flag)
```

flag{n1c3_j0b}