

2018 第十一届全国大学生信息安全竞赛 逆向RE writeup

原创

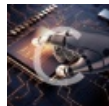
delia0204 于 2018-05-04 21:01:21 发布 5462 收藏 5

分类专栏: [CTF](#)

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/anastasia0204/article/details/80200169>

版权



[CTF 专栏收录该内容](#)

20 篇文章 0 订阅

订阅专栏

题目:

看似一个check, check都通过之后success

```
while ( v13 != 3 ); // flag1 flag2 flag3
if ( (unsigned int)check1(__flag1) )
    return 0xFFFFFFFFLL;
if ( (unsigned int)check2(flag2) )
    return 0xFFFFFFFFLL;
if ( (unsigned int)check3(flag3) )
    return 0xFFFFFFFFLL;
puts("Congratulations!");
```

解题:

逆向思路-----逆着写算法, 发现第三块的md5没有解, 解出第三块flag内容需要其他后门, pass

爆破-----发现flag123块的长度不定只知道是0-16, 这样爆破是不可行的, pass

other way solve 第三块flag

在check3 中check通过之后会写flag文件, 看起来是个道道, , ,

```
result = 0xFFFFFFFFLL;
if ( (!v6 && !v7) == v6 )
{
    v12 = fopen("flag", "w+");
    if ( v12 )
    {
        v13 = &qword_6038A0;
        v14 = 0;
        do
        {
            v14 += *(unsigned __int8 *)v13;
            v13 = (__int64 *)((char *)v13 + 1);
        }
        while ( &qword_6038B0 != v13 );
        v15 = _flag3[3] ^ (v14 >> 4);
        v16 = _flag3[4] ^ v14 & 0xF;
        v17 = 0LL;
        do // write flag
        {
            if ( v17 & 1 )
                byte_6020E0[v17] ^= v16; // ji byte
            else
                byte_6020E0[v17] ^= v15; // ou byte
            ++v17;
        }
        while ( v17 != 6047 );
        fwrite(byte_6020E0, 0x179FULL, 1uLL, v12);
        fclose(v12);
    }
}
```

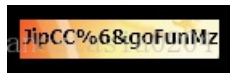
写flag文件需要的参数 (byte_6020E0, (char)v16, (char)v15), 其中byte_6020E0是已知的, 爆破v16, v15两个8字节即可。

爆破把flag内容写在一起有390M,到哪里找flag。。。。

这个坑，进去之后觉得自己很是凉凉

最终想到了这是个文件！啊！，可以根据文件头找信息啊，常见的文件类型首先想到了图片，又想到了JPG(头 FF D8 FF E0 00 10 4A 46),果然有!!!

拿到flag



附件：

题目，ida数据，逆向用到的数据，所有脚本上传到了我的资源文件里边