

逆向-攻防世界-notsequence

转载

山清水秀iOS 于 2019-06-11 22:54:00 发布 107 收藏

原文链接: <http://www.cnblogs.com/whitehawk/p/11006824.html>

版权
越来越喜欢逆向了, 通过调试解决程序的逻辑问题真是太爽了。后天单片机考试, 全球黑客保佑孩子不挂科。
。。。

看看这道题, 自己算法太菜了。

没有壳, 直接IDA吧。

```
memset(&unk_8049BE0, 0, 0x4000u);
puts("input raw_flag please:");
v3 = &unk_8049BE0;
do
{
    v0 = v3;
    ++v3;
    scanf("%d", v0);
}
while ( *(v3 - 1) != 0 );
v2 = sub_80486CD((int)&unk_8049BE0);
if ( v2 == -1 )
{
    printf("check1 not pass");
    system("pause");
}
if ( (unsigned __int8)sub_8048783((int)&unk_8049BE0, v2) ^ 1 )
{
    printf("check2 not pass!");
    exit(0);
}
if ( v2 == 20 )
{
    puts("Congratulations! fl4g is :\nRCTF{md5(/*what you input without space or \\n~/)}");
    exit(0);
}
```

输入, 相当于一个数组吧, 每次输入一个值, 输入的不是数字好像就结束了。

两个check

然后是得到flag, 就是MD5 (正确的输入的值)

分析check1.

```
7
8 v5 = 0;
9 for ( for__i = 0; for__i <= 1024 && *(_DWORD *) (4 * for__i + a1); for__i = v5 * (v5 + 1) / 2 )
0 {
1     v3 = 0;
2     for ( i = 0; i <= v5; ++i )
3         v3 += *(_DWORD *) (4 * (i + for__i) + a1);
4     if ( 1 << v5 != v3 )
5         return -1;
6     ++v5;
7 }
8 return v5;
9
```

根据程序, 列出了几个数, 1 11 112 1133, 如果acm不错的话, 这道题已经出来了, 然而我是个垃圾。

然后看check2,

```
v6 = 0;
for ( i = 1; i < a2; ++i )
{
    v4 = 0;
    v3 = i - 1;
    if ( !*( _DWORD * )( 4 * i + a1 ) )
        return 0;
    while ( a2 - 1 > v3 )
    {
        v4 += *( _DWORD * )( 4 * ( v3 * ( v3 + 1 ) / 2 + v6 ) + a1 );
        ++v3;
    }
    if ( *( _DWORD * )( 4 * ( v3 * ( v3 + 1 ) / 2 + i ) + a1 ) != v4 )
        return 0;
    ++v6;
}
return 1;
}
```

看起来是将位置1,3,6,10,15,等的值加起来,然后和最后一个位置的后一个数比较。我擦,我不会我不会。看了writeup,牛逼啊,杨辉三角,杨辉师傅真强。呵呵哈哈。

1 11 112 1133我们的这组数应该写成, 1 11 121 1331,确实是杨辉三角,网上搜了下python的杨辉三角,有个师傅真是tql,你们去学习下人家的代码吧,脑子真是个好东西,脑子不好,必须得努力努力再努力。给你们网址, <https://blog.csdn.net/zyz1431/article/details/79104035>,这里就不贴师傅的代码了,去给师傅涨涨人气吧。废话好多。。。没有女朋友,自己跟自己聊天,哈哈。

```
if ( v2 == 20 )
{
    puts("Congratulations! fl4g is :\nRCTF{md5(/*what you input without space or \\n~/)}");
    exit(0).
}
```

20说明是20行,也就是210个元素。

拿到杨辉三角,去掉空格和换行,MD5加密即可。格式是RCTF{}

转载于:<https://www.cnblogs.com/whitehawk/p/11006824.html>