170524 逆向-CrackMe(1)

原创

奈沙夜影 于 2017-05-24 21:52:15 发布 658 收藏
分类专栏： CrackMe

本文链接：https://blog.csdn.net/whklhhhh/article/details/72700446


CrackMe 专栏收录该内容
83 篇文章 2 订阅
订阅专栏

1625-5 王子昂 总结《2017年5月24日》【连续第235天总结】

A.CrackMe练习 01

B.Acid burn 看了一些论坛上的crackme，都无从下手

找到了这个练习包，参考了一些writeup，终于能动手尝试了

打开首先是一个nag窗口，刚开始我还没反应过来，后来看writeup的时候才知道这个窗口需要屏蔽掉（明明标题上写着please kill

首先进行字符串搜索，查找相关的字符“welcome”，直接就出来了

跟踪到此处，发现先是push 0，然后将各个字符串的内存地址赋给寄存器，再call一个函数

这不就是用寄存器传递参数的方法嘛

直接将call的指令nop掉

然后报错了_(:3」∠)_地址冲突

想了一下，应该是由于约定中函数负责清理栈堆，call之前的准备工作中，寄存器的赋值没有影响，但是Push 0对栈堆造成了影

而NOP掉call之后，这个0没有人清理了

因此把push 0也nop掉，终于没问题

进入程序，有两个部分“Name/Serial"和"Serial"。分别是ID-序列号和验证码两个题

先从简单的验证码入手

点开以后直接随便输入，弹出“Try again!"

字符串搜索后找到相关指令

下断点，然后运行

注意观察着寄存器，在下文有正确的提示“Congratz!"，之间有一个call和jnz

那么想必就是call进行判断，jnz进行跳转了吧

在前面一些下断点，发现运行到call的时候，寄存器中被放入了指向字符串的内存地址

EAX处是我随便输入的“111”，EDX中则是“Hello Dude!"

为了确认，跟进call

发现call里面就是对EAX和EDX的cmp，因此可以确定正确验证码就是直接保存在内存中的“Hello Dude!"了

至此，爆破和正确验证码都得到了

最难的Name/Serial比较复杂

爆破起来倒是没有难度，直接搜索字符串，找到对应的jmp，nop掉即可

难的是找到找出序列号的算法写出注册机

一步一步观察栈堆和寄存器的情况，最终锁定在一个call中

这个call跟进去却特别复杂，无从下手

尝试了很久都得不到解

最后查找了网上的多个writeup，最终明白：

算法是在最外面：

首先判断字符数，小于4直接failed

取第一个字符，乘以29再乘以2，得到的数就是序列号中间的数了

很复杂的那个call是数字转换字符串的功能，最后再拼接上"CW-"和"-CRACKED"

局部变量的部分还需要好好学习一下

有一个思路，从getwindowtext的API跟到获取输入字符串的地方，然后再一步一步跟踪处理过程，应该能更清晰一点

然而并没有找到这个API在其中的调用位置_(:3」∠)_

大概是学艺不精吧，以后再试试

C.明日计划

crackme-02