170531 逆向-CrackMe之007

原创

奈沙夜影 于 2017-05-31 23:43:04 发布 590 收藏
分类专栏： CrackMe

本文链接：https://blog.csdn.net/whklhhhh/article/details/72824942


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31日》【连续第242天总结】

A.CrackMe（7）

B.这次挑战一下2星的程序

拖入PEiD，无壳，Delphi编写的。

跟昨天的是同一个作者，同样分多关、隐藏按钮的套路

不过这次OK是直接显示的，虽然也有Cancella

这次比昨天的难，反而有文本字符串了，直接搜索就能找到相应位置

不过有好用的Dede干嘛不用呢，还能方便地直接标出API的名字

拖入Dede，一共有三个事件，分别响应cancella、register和again

again还不知道是啥，按照昨天的顺序，先跟cancella试试吧

找到事件，往下翻几步就看到了熟悉的test ax,ax和jz 这里大概就是关键跳转了，爆破之

弹出了两个框，“GREAT！"和”LAMER“

前者很明显是恭喜的意思，但是LAMER我百度了一下，除了最多的是一个服装品牌，百度翻译说是蹩脚的、差劲的意思

嗯？而且成功以后只有弹窗，没有任何变化

虽然很奇怪，但是上上下下都翻了几遍也没看到别的路径了只好先这样

同样，找到register的地址，判断方法相同，爆破后register按钮消失，出现了"...again"按钮

此时Name的框变成了灰色无法输入状态，看样子两次的Name是不允许改变的啊

同样，在Dede中找到again事件的地址，爆破，全部消失露出了Logo。

值得一提的是，因为register存在内层的一个关键call，我刚开始爆破内部call的时候，后面尝试点击again居然也直接通过了

后面才发现again事件和register事件代码完全一样，调用的是同一个函数（call）

算法：

先是cancella

跟进去有两次关键跳转，跟昨天相似

最外部通过test ax,ax来判断是否为0；然后第二层再一个test

跟到关键call里面以后，进行了一番非常复杂的计算。但是核心的只有长度比较和字符串ASCII比较

只要取两个字符相同的、长度都大于4的字符串即可

输入后反应与爆破相同

然后看register

跟进去，在关键call中耐着性子读每行代码

首先是一个复杂的双层嵌套循环，对每个字符，都与整个字符串的所有ASCII码相乘一遍，然后再乘以一个数，最后全部累加起

求和结果与0做异或运算，应该是取反的效果

修正：与一个小伙伴讨论，又去查找了一下，该段代码应为abs绝对值函数的优化汇编

若为正数，dx为0，xor 0和sub 0的结果都不变

若为负数，dx为-1，xor -1为取反，sub -1为+1，即求该负数的补码，即求相反数

综上是为绝对值函数

（原代码为mov eax,ebx

cdq

xoreax,edx

subeax,edx)

感觉在bx作为累加和恒为正的前提下，dx一定为0，则做异或运算和作差的结果应该就是取反吧

再除以0xA2C2A，保留余数

然后拿出code转换成整型的值，用除以0x59的商+除以0x50的余数+1，与之前的数比较，相等则通过

之前说了Name的最终求和为0，最后的余数也为0。而Code的结果有一个+1，除非是造成溢出才有可能得到0。然而作为32位除

向上跟踪，发现这个数的路径是ds:[0x445830]→eax→edi，跟到0x445830的地址，附近全是空的，压根不在数据存储附近

有点懵逼了，上下再看看代码，突然发现之前还有一处出现了mov ds:[0x445830],eax的语句。这不就是我梦寐以求的来源吗，赶

那还等什么，输一个字母再跟进去呗：首先长度要大于5，然后取每个字符的ASCII除以0x11，余数+1后乘上前一个字符的ASC

总结一下，在这个算法中，一个Name需要搭配两次Code,分别为字母A和数字N；最终需要满足f3[f1(Name) , f2(A)]=f4(N)的关系

again和register是一样的算法，将之前得到的数再输入一遍即可

之后看了一下之前参考的52破解论坛的CrackMe系列的writeup，发现他没有找到0x445830的位置，就卡着了┐(ﾟ_ﾟ)┌

不过后面跟帖里有人给出了正解，跟我想的差不多

C.明日计划

CrackMe(8)