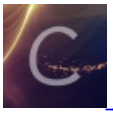
【reversing.kr逆向之旅】Replace的writeup

原创

iqiqiya 于 2018-11-10 11:26:47 发布 642 收藏
分类专栏： 我的逆向之路 ------reversing.kr 我的CTF之路 文章标签： Replace reversing.kr逆向 writeup reversing.kr

本文链接：https://blog.csdn.net/xiangshangbashaonian/article/details/83927224


我的逆向之路 同时被 3 个专栏收录
108 篇文章 10 订阅
订阅专栏

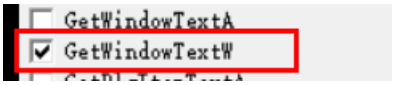

------reversing.kr
11 篇文章 0 订阅
订阅专栏


我的CTF之路
92 篇文章 5 订阅
订阅专栏

无壳 vc++程序

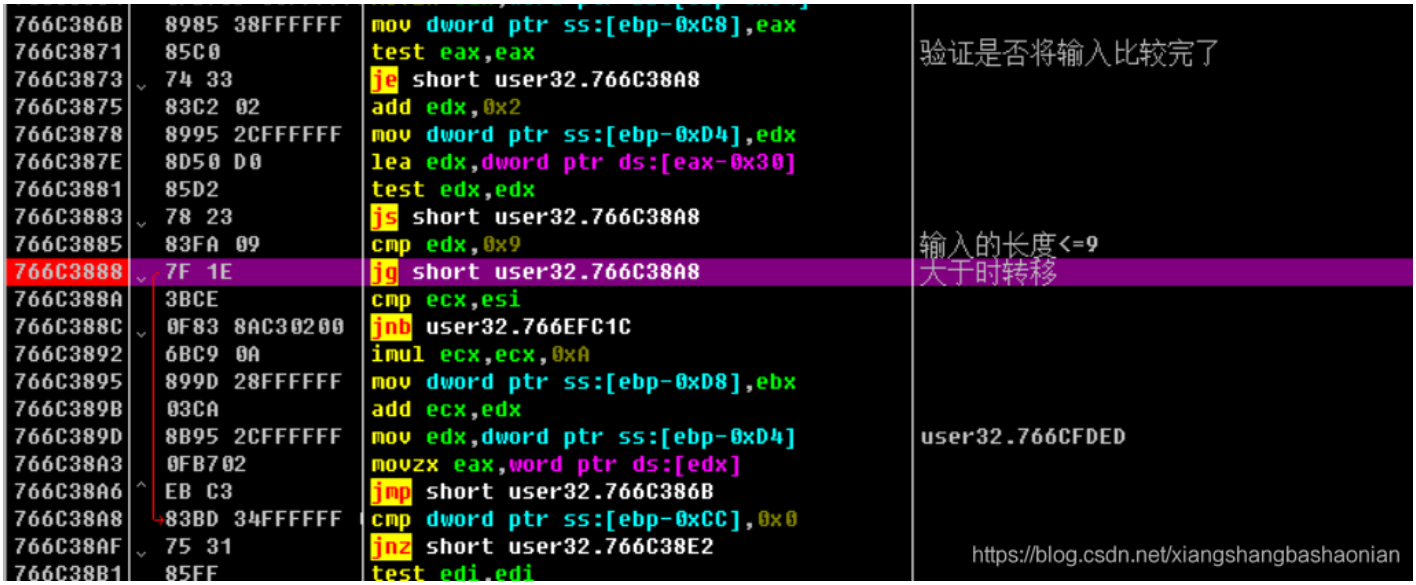
载入Olydbg动态调试分析就好

可以通过下API断点GetWindowTextW来找到关键代码



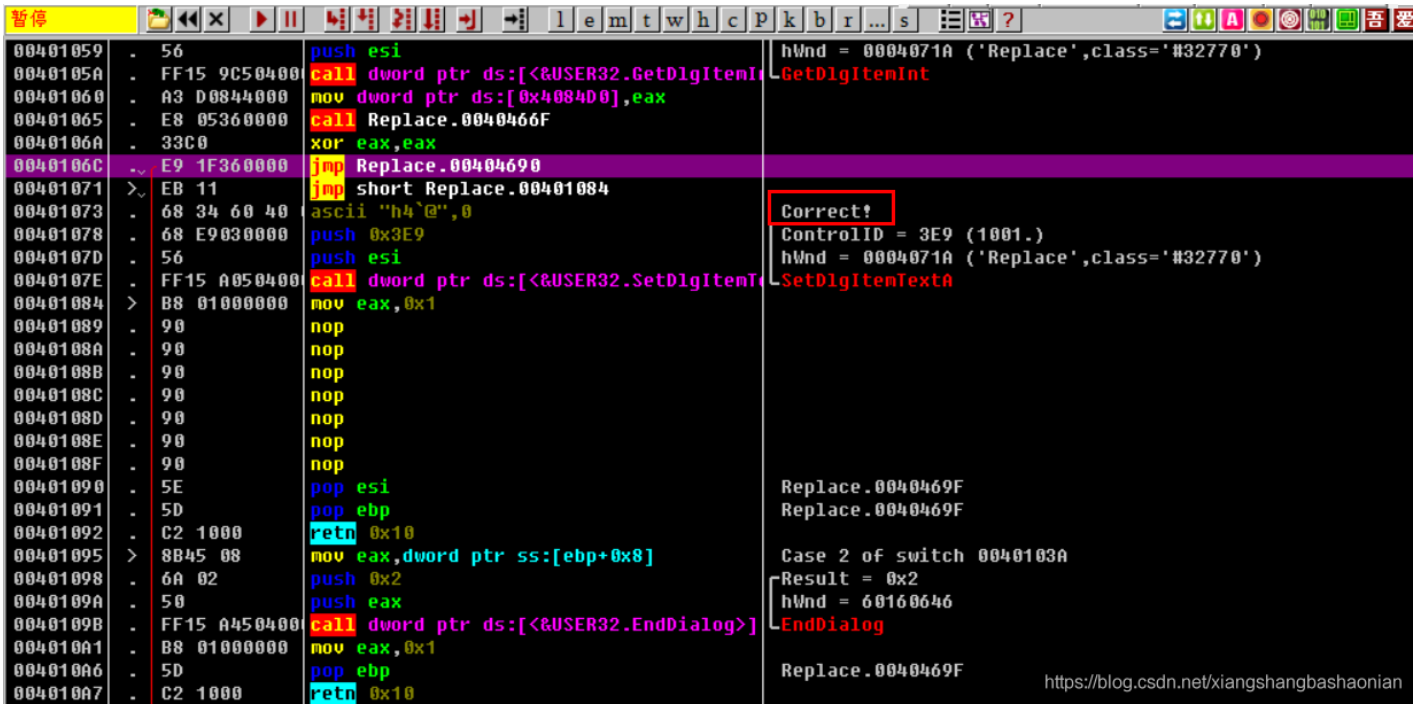
因为它就是用来捕获我们的输入的 随意输入(这里可以发现只可以输入数字 长度什么的倒是没有限制)

就像下面一样 可以看到地址很大 在系统领空 user32.dll 所以这里不用管



上面循环过去之后

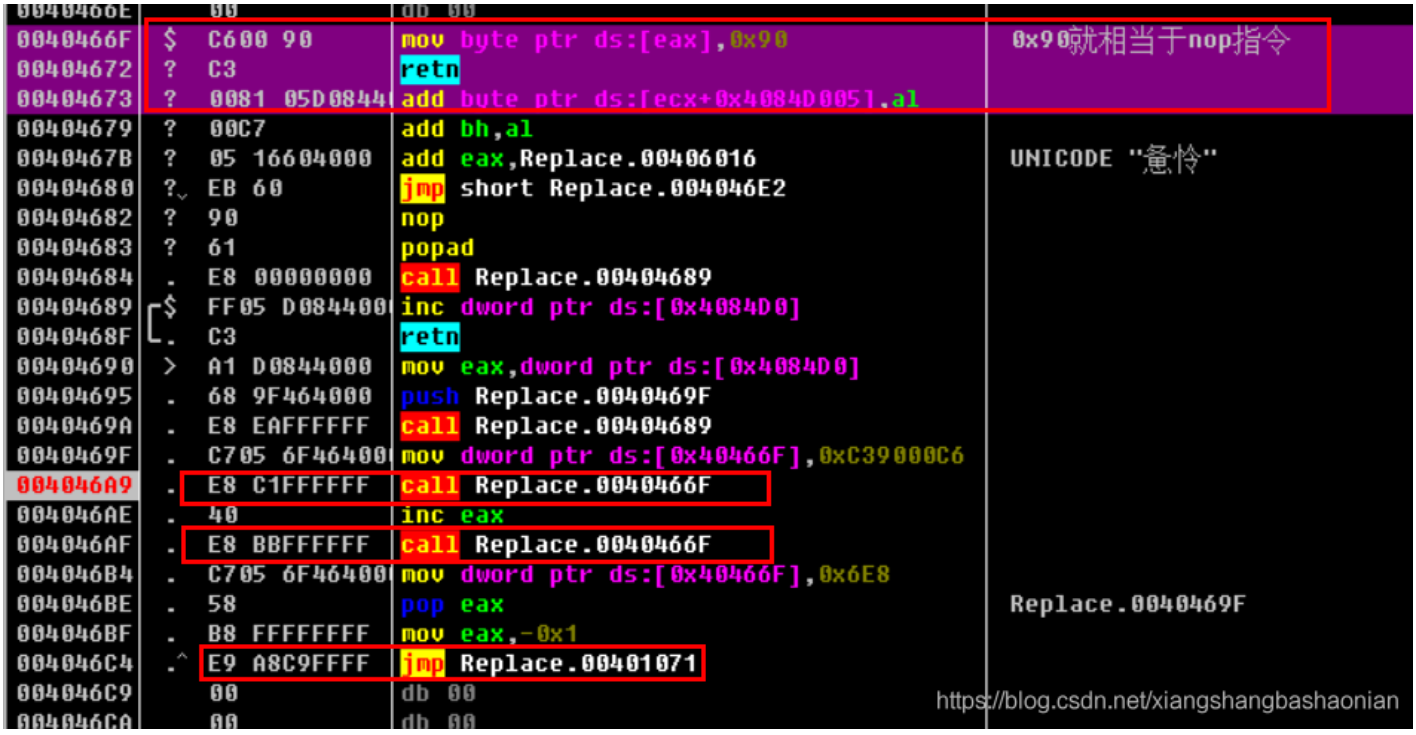
就来到了这里



可以看到成功的标志Correct!

但是CPU窗口有两个jmp指令 也就是说无条件跳转 肯定无法成功 向上向下跟踪 可以看到只有地址00401071处有一个跳转进来 可以使我们走向成功

接着向下单步 来到下图所示



可以发现 程序不管输入什么 总是走到0x004046A9这里就停止运行了

可以看到这个call跳转向的地址 是将eax所对应的地址进行nop

连续两个这样的call 中间inc eax 就是eax加1 也就是说连续nop两个地址

再来看下边那个向上跳转的jmp 跳转的地址0x00401071 对应其中一个jmp 挨着的下一条指令正好是个空指令

那么我们就可以利用这两个call 将这两条指令进行nop 即可走向Correct!

这时候重点就在eax是如何与我们的输入进行运算的

输入123时 eax=0x60160646 把123转成十六进制是0x7b 0x60160646-0x7b=0x601605cb

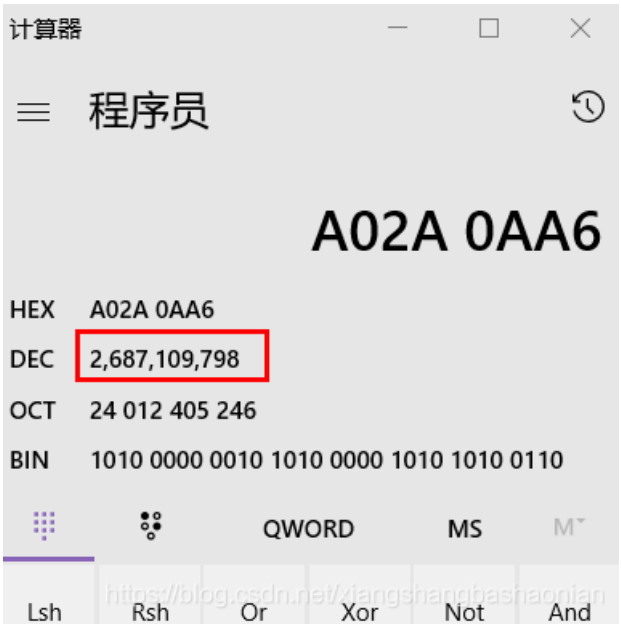
重新输入1234时 eax=0x60160A9D 把1234转成十六进制是0x4d2 0x60160A9D-0x4d2=0x601605cb

多次输入 可以发现最终结果都是0x601605cb

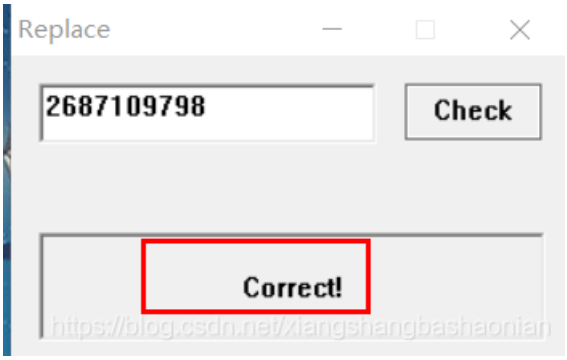
那么我们正确的输入应该要满足0x00401071-?=0x601605cb

可以发现需要eax溢出才可能成立

所以？= 0x100401071-0x601605cb = 0xa02a0aa6



验证2687109798 成功



参考链接：

http://www.mottoin.com/article/reverse/88447.html

https://www.cnblogs.com/xiao-zhang/articles/5602504.html