实验一 Afkayas1.exe 程序的逆向分析

1180300829 余涛

一. 实验目的

1. 掌握逆向分析的一般流程，熟练使用逆向分析的常用工具，并给出实验中相应软件的输出结果；
2. 掌握逆向分析的断点设置方法并对关键程序逻辑进行跟踪和定位；
3. 掌握逆向分析的指令修改方法，对程序的序列号验证机制进行爆破；
4. 掌握常用的汇编指令，对程序的序列号算法进行回溯。

二. 实验内容

1. 第一步：查看程序启动情况

(1)找到合适的查壳工具完成对 Afkayas1.exe 程序的查壳，判断程序是否有壳和壳的类型，截图如下（图 1）：

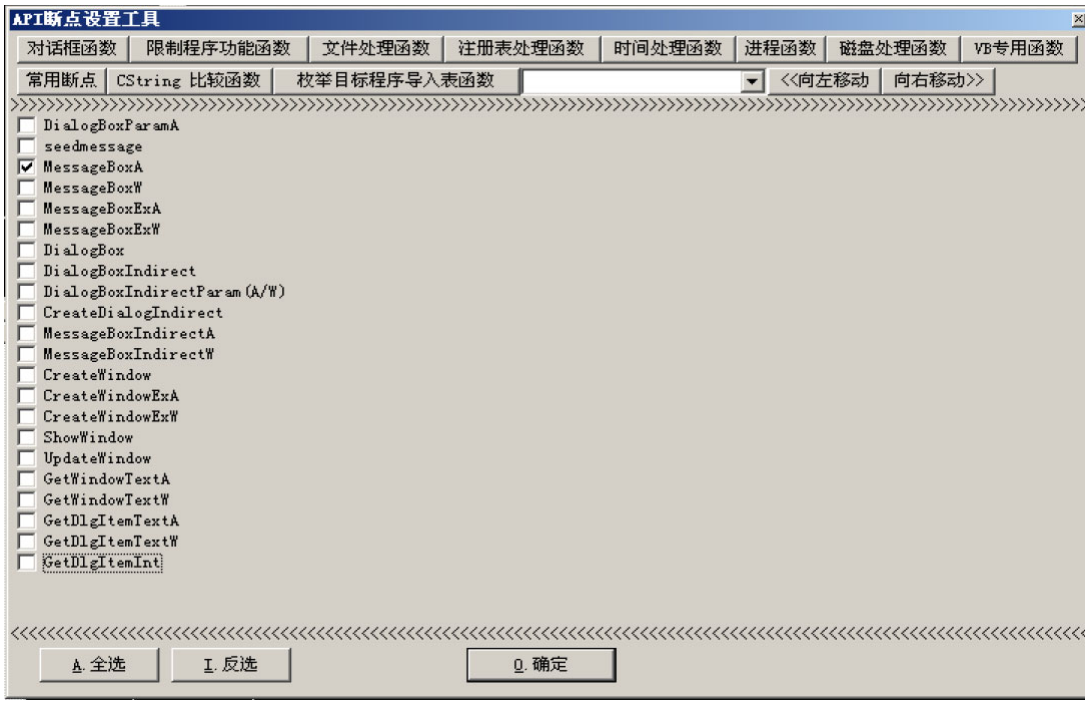
无壳

(2)启动 Ollydbg，加载程序；

(3)输入学号和序列号之后会出现提示窗口，截图如下（图 2）：

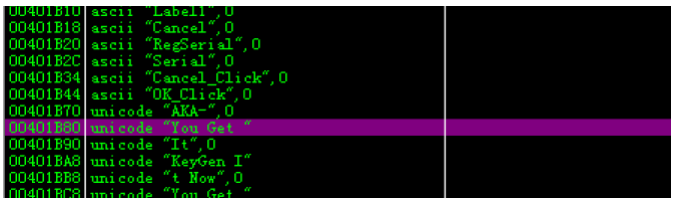
2. 第二步：爆破法 Crack 得到正确提示窗口

(1)弹窗是通过调用 MessageBoxA 实现的，请在 Ollydbg 中采用 API 断点设置工具 方式为此函数设置断点，截图如下（图 3）：



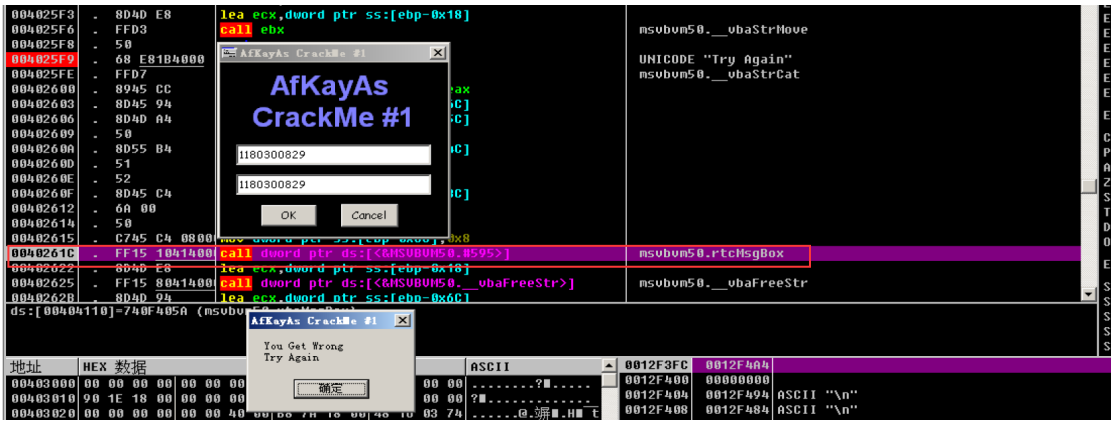
(2)执行程序发现断点并未成功，寻找其他的突破点。在第一步出现

的验证窗口中，出现了字符串“You Get Wrong Try Again”，因此，可以以字符串作为切入点尝试。在 Ollydbg 中点击右键，点击查找-所有参考文本字符串。在新的文本字符串列表界面里，点击右键，查找，输入文本“You Get”，选择区分大小写，找到其出现的地址，截图如下（图 4）：

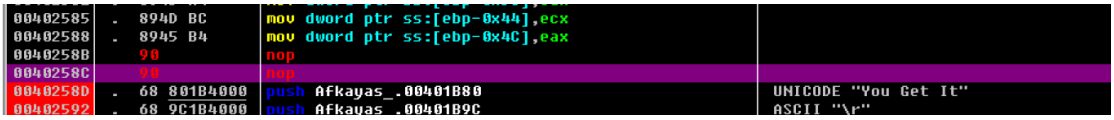


地址为 00401B80

(3)右键选中该条，点击在每个指令上设置断点。重新执行程序，可以看到中断在验证窗口弹出之前，向下单步执行，可以看到地址为 00402533 处调用了__vbaStrCmp 函数。继续向下单步执行，在地址为 0040258B 处发生跳转，跳转指令为 je 指令，跳转到的位置出现了验证窗口的相关内容和函数调用，窗口出现在调用了 msvbvm50.rtcMsgBox 函数之后，该段指令截图如下（图 5）：



(4)从而可以得知，程序调用__vbaStrCmp 函数，比较输入序列号和正确序列号，由比较结果决定是否跳转，弹出正确或错误的提示窗口。那么，可以简单地用 nop 指令代替跳转指令，不让跳转发生。

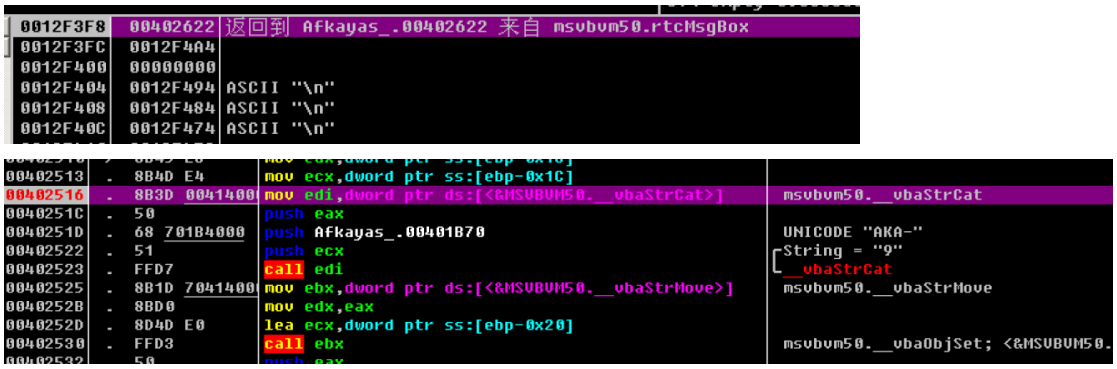


(5)点击菜单中的“b”按钮，将所有断点禁止。重新执行程序，（注：重新开始程序后，需要按 Ctrl+P 快捷键或点击菜单中的“P 按钮”打开 patches 窗口，发现状态显示“已删除”，即已经修改的指令被禁用，需要选中被删除的 Patch 按空格让状态变为“激活”，然后再运行程序。）输入学号和任意序列号，点击“OK”，得到正确的提示窗口，截图如下（图 6）：

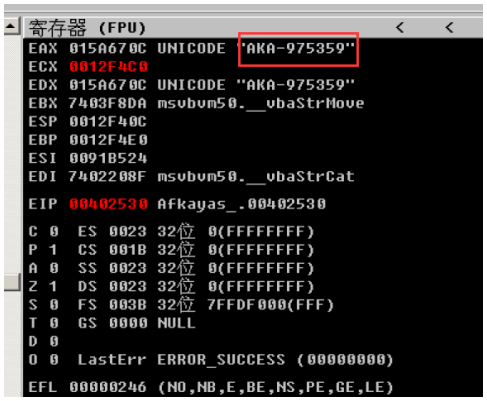
3. **第三步：爆破法 Crack 得到正确序列号**

(1)在第二步中，观察到窗口在调用第二步(3)提到的函数之后出现，

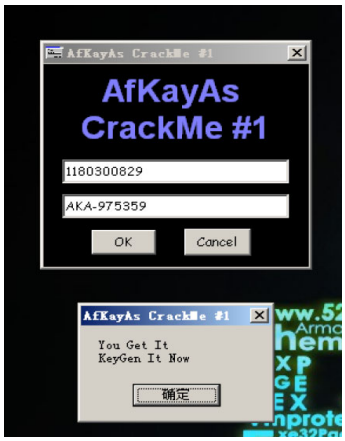
可以在命令行输入指令 bpx 函数名 对该函数下断点。重新执行程序到断点处，可以看到，此时栈顶值为 00402622，这是该函数的返回地址。选中它使用回车键回到调用者层，往上可以看到，在__vbaStrCmp 函数调用前，地址为 00402516 处调用__vbaStrCat 函数，这说明，正确序列号的产生可能存在拼接操作。



(2)在该地址下断点，重新执行程序，输入学号和任意序列号，中断时寄存器 ecx 的 UNICODE 值为 975359，单步步过之后，可以看到寄存器 eax 中保存着正确序列号的 UNICODE 值，为 AKA-975359，向下执行到__vbaStrCmp 函数的 call 指令，可以看到栈顶已压入两个参数，输入序列号和正确序列号。继续向下，可以看到程序根据比较结果进行了跳转，这符合前面的猜测。



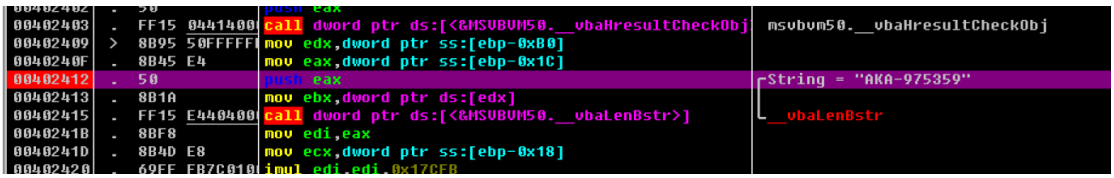
(3)那么根据上述__vbaStrCat 执行完毕得到的正确序列号，将所有断点禁止，重新执行程序，输入学号和正确序列号，最终将弹出正确的提示窗口。



4. **第四步：序列号算法分析**

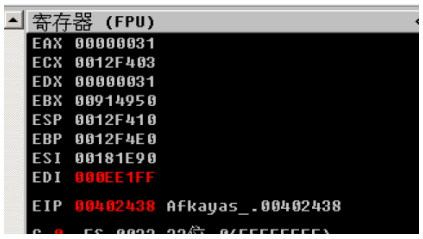
(1)第三步中，已经成功得到了正确序列号，但在其拼接之前，寄存器 ecx 中的值是如何产生的？接下来将对序列号中的数字部分的产生算法进行分析。

(2)保留第三步的断点，重新执行程序，同样地通过栈顶返回到调用层，一直上移可以看到在地址 0x402412 处附近开始有字符串的相关操作，因此在该地址下断点。



(3)重新执行程序，程序中断在该断点处，此时寄存器 eax 中存放着输入的学号，单步步过，程序调用__vbaLenBstr 函数，获取了输入学号的字符长度，将结果存放在寄存器 eax 中。随后，程序将该值移动到寄存器 edi 中，并与值 0X17CFB 进行相乘运算。

(4)然后，程序调用 rtcAnsiValueBstr 函数获取输入学号的第一个字符的 ASCII 码，然后将 ASCII 码与寄存器 edi 中的值相加，得到值为 00EE1FF 。



(5)进一步地，程序调用__vbaStrI4 函数，将寄存器 edi 中的十六进制值转换为十进制值的字符串形式，可以看到，此时的字符串正是正确序列号的数字部分。

三. 思考题

1. 逆向分析过程中，有着不同的关键逻辑汇编代码定位方法和不同汇编指令方式改变关键逻辑，描述至少一种其他方式.

答：（1）有条件跳转修改为无条件跳转。

（2）将函数调用 call 修改为空指令 nop。

2. 为什么对 MessageBoxA 下断点不成功，而对 rtcMsgBox 下断点成功了？

答:程序一直在调用 rtcMsgBox 方法,而没有调用过 MessageBoxA，所以对 rtcMsgBox 下断点成功。

3. 根据上述的算法分析，简述序列号的完整生成算法.

答：序列号都具有 AKA-XXXXX 的形式，所以需要获得 XXXXX，

先输入字符串的长度，将长度和 0x17CFB 进行有符号乘法，得到结果 A。然后获取字符串第一个字符的 ASCII 码，对该 ASCII 码与 A 相加，这样就得到了序列号。