**前言**

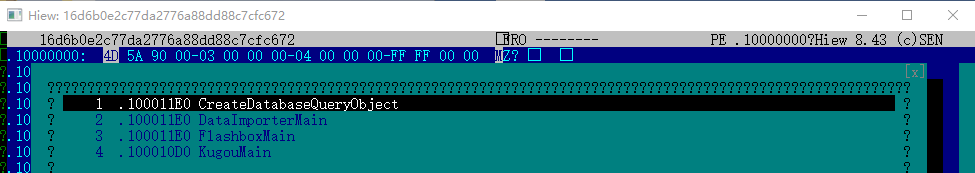
我们本次课程以及下次课程所讲解的是PE类恶意文件的快速分析技巧。事实上，与我们之前所讲述的脚本类以及文档类恶意程序相比，PE类恶意程序可以说是最需要技巧与经验才能够得到正确分析结果的一类恶意程序了。而这类恶意程序的种类与数量也可以说是最多的，免杀手段也是五花八门的，所以一般来说，分析PE类恶意程序是需要非常多的分析工具作为辅助的。但是在我们目前的基础篇中，我在这里为大家选择的一些PE类恶意程序，其实都是有着一些特点的，凭着这些特点，我们就能够实现在仅仅利用Hiew的前提下，进行快速分析。当然，这里给大家总结出来的技巧，也是我在实际的分析工作中所总结出来的，因此具有很强的实践指导意义，希望大家掌握。

**样本分析**

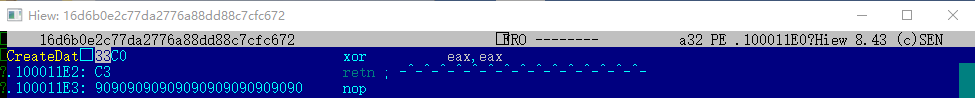
PE类程序指的是基于Windows操作系统的可执行程序，最常见的有exe程序、dll程序以及sys程序，不同类的PE程序有着不同的应用范围，从而也就导致了PE类恶意程序的表现形式的多样性，分析不同类型的PE类恶意程序所采取的技巧方式也是有所不同的。比如我们这次的这些程序：

**1、16d6b0e2c77da2776a88dd88c7cfc672**

使用Hiew打开这个样本之后，首先由程序开头部分的MZ与PE就可以确定这是一个PE类的程序，我们按一下回车，切换到十六进制分析模式，可以看到它的地址是以0x10000000作为开始的，那么以这种地址作为开始的文件，基本上就是属于DLL文件了。接下来我们按一下F8，也就是查看一下这个文件的头部信息。可以看到在弹出相应信息的同时，Hiew下方的快捷按钮也产生了变化，这里出现了一个“9Export”，意思是按下F9就可以查看文件的导出表，这个导出表可以理解为DLL程序的功能部分，我们不妨查看一下：



可以看到，这个DLL程序包含有四个导出函数，其中的第四个函数名称是KugouMain，从名称上来看，这个DLL文件似乎是酷狗的一个组件。可奇怪的是，前三个函数的地址竟然都是0x100011E0。要知道，对于一个DLL程序而言，它的功能可以说都包含在导出函数中了，那么不同的导出函数为了实现不同的功能，势必会有不同的代码，同时这些代码一定也会保存在不同的地址位置。如果三个导出函数都共用一个地址的话，那么可以理解为，我不管调用这三个导出函数其中的哪个函数，所执行的其实都是相同的功能，正常的DLL程序是绝对不会这么干的。如果我们进入函数的内部查看它们的代码，可以看到如下内容：

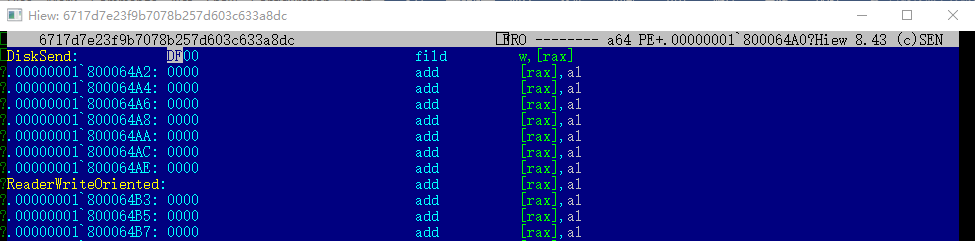


这里仅仅是对eax寄存器自身进行异或运算，也就是清零操作，然后接下来就retn返回了，可以认为这个函数并没有执行什么实际的功能，也就是说，酷狗程序不可能会写这种没用的函数，那么也就可以确定，这是一个有问题的程序了。

实际上，对于这种将自身伪装成大公司编写的程序组件的恶意程序，我们都可以把它们归类为“DLL劫持”或者“白加黑”类的恶意程序，因此可以把它命名为：Trojan.Win32.Dllhijack.a。

**2、6717d7e23f9b7078b257d603c633a8dc**

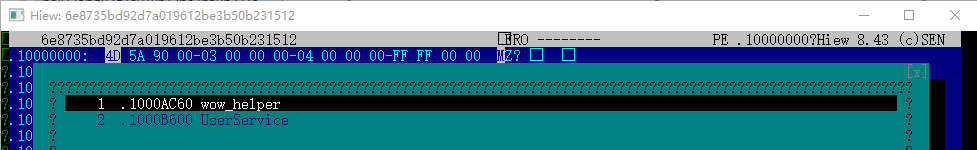
首先由MZ和PE可以判定它是一个PE类的程序，然后按下回车，在十六进制界面中，可以发现它的起始地址是0x00000001~80000000，也就是说，这个程序是一个64位的程序，按下F8发现它有导出表，按下F9查看导出表，可以发现它包含有两个导出函数，这里不妨选择第一个导出函数查看一下：



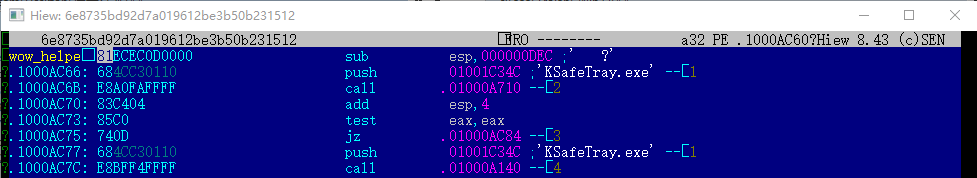
可以看到，这两个导出函数的代码内容基本上都是00，也就是并未执行实际的动作，正常的DLL程序是不可能出现这样的情况，也就是什么都不做的，所以现在就可以将其判定为是恶意程序了，所以这里可以将其命名为Trojan.Win64.Agent.a。

**3、6e8735bd92d7a019612be3b50b231512**

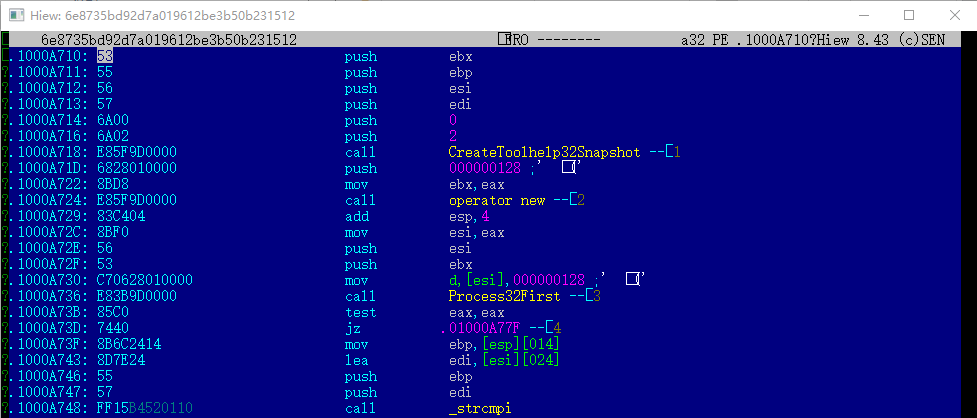
对于这个程序，我们同样可以发现它是一个DLL程序，包含有两个导出函数：



不妨进入第一个导出函数查看一下：

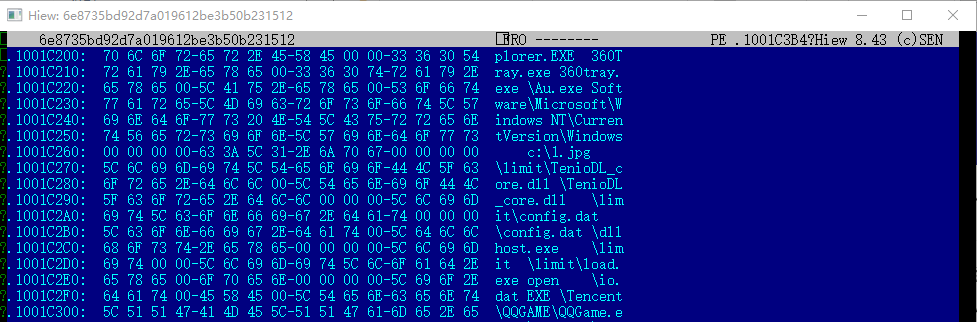


可以看到，这里出现了KSafeTray.exe这个名称，它是金山安全软件的一个组件。其实这里出现这个名称是一件很奇怪的事情，因为一个正常软件里面，不应该出现一个杀软厂商的文件名或者进程名。如果分析一下这段程序，可以看到它是利用push语句将这个字符压栈，作为接下来的call的一个参数，那么我们不妨进入这个call的内部查看一下，看看它利用KSafeTray.exe做了什么，这里可以直接按数字2，跳转到call的代码位置：



可以看到，这个call主要执行的操作就是进程的遍历，那么也就可以理解为，这个程序实际上就是在当前进程中查找是否包含有KSafeTray.exe这个进程，那么这就是一件很奇怪的事情了，正常程序是没理由这么干的，只有恶意程序，才可能会采用这种方式，遍历进程中的安全软件进程，然后将其关闭，或者是其它的一些恶意操作。

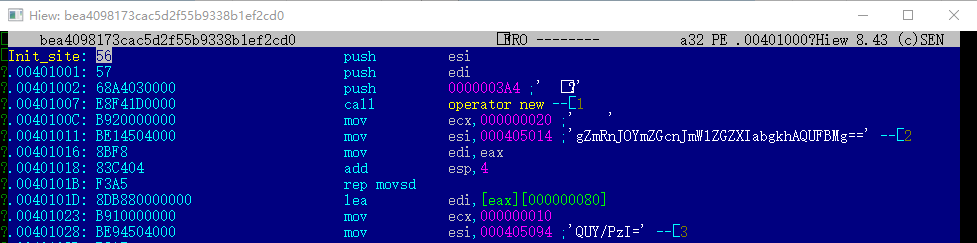
弄清楚了这个call的大概功能之后，我们可以按退格键（Backspace）退回到上一个界面。刚才我们是按下数字2，跳转到了call函数的代码代码位置，这里我们不妨按下数字1，看看“KSafeTray.exe” 这个字符串存储位置附近都有什么：



在这里，我们还发现了“360Tray.exe”以及“QQGAME”等字符，也是正常程序中不应该同时出现的，那么由此就可以断定说，它就是一个恶意程序了。最后，可以将其命名为Backdoor.Win32.Agent.a。

**4、bea4098173cac5d2f55b9338b1ef2cd0**

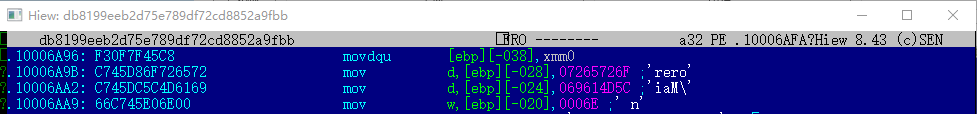
首先，这个程序依旧是一个DLL程序，并且只有一个导出函数，进入函数内部查看一下代码：



可以看到，程序中出现了大量的以BASE64加密的内容，这也是一个正常程序中不应该出现的，这种形式类似于我们之前分析过的脚本类以及文档类恶意程序，所以就可以直接将其报毒并进行归类了：Backdoor.Win32.Agent.b。

**5、db8199eeb2d75e789df72cd8852a9fbb**

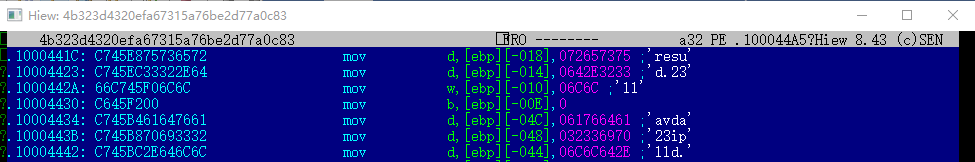
这个程序是一个DLL程序，并且有两个导出函数，由于这两个导出函数的地址是相同的，因此可以直接断定它是一个有问题的程序了。或者我们也可以查看一下它的代码：



可以看到，在这段代码中，它是将字符“Main”拆分，利用ASCII码的形式，一个一个地放入内存中，而不是我们常见的那样，将一个完整字符串保存在某一个地址中，然后调用该地址，从而获取字符串。综合以上，可以将其分类为：Rootkit.Win32.blackken.a。

这里所提到的将字符逐个放入内存的方式，从而组成完整字符串的方法，实际上在当前的恶意程序或者是一些灰色程序（即not-a-virus类程序，比如广告程序）中是经常会遇到的。其实，对于这种特征鲜明的程序，我们如果再使用传统的特征提取的方法来对抗这类程序，就显得力不从心了。因此有些知名杀毒厂商会采用启发的方式来应对这类威胁，对于这种技术的原理，在未来的高级篇中，会给大家详细介绍。

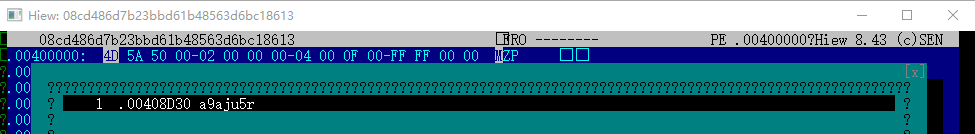
类似的程序还有**4b323d4320efa67315a76be2d77a0c83**，这个程序也是一个DLL程序，并且还有很多的导出函数，我们重点看一下位于0x10004258的名为DownloadFile的这个导出函数：



可以看到，这里大量采用了上述提到的方法，因此可以直接给它归类为：Rootkit.Win32.blackken.b。

**6、08cd486d7b23bbd61b48563d6bc18613**

查看这个文件的导出表，可以看到：



我们发现，这个程序只有一个导出函数，函数名为a9aju5r，很显然，正常的程序并不会起一个这样的名字。有理由怀疑，可能是恶意程序的作者使用了某种自动化生成恶意程序的工具，为了使得每个恶意程序不相同，于是就设置它们的导出函数名称各不相同，从而避免杀软的初步查杀。于是我们可以直接将其归类为：Trojan.Win32.Agent.a。

**小结**

本次课程所讲述的就是我们在日常分析的工作中最常遇到的DLL恶意程序的表现形式，其核心原则依旧是找到它与正常程序的不同，希望大家掌握。