**Ring3下注入DLL的另类方法，能过杀软和游戏NP（源码）**

**注入DLL是做全局钩子或者拦截类软件都有可能用到的技术，如果做外挂的话我们也有可能需要注入一个DLL到游戏进程中去干点什么“坏事”。 但我们知道现在要注入DLL是越来越难了。**

**场景1：制作火星文输入法外挂，原理是利用API HOOK拦截并修改输入法相关函数，需要注入一个DLL到所有进程中，但是后来发现，在开启了瑞星的帐号保险箱后，用户将不能在QQ中输入火星文。原因是瑞星保护了QQ进程，禁止对其注入DLL，解决方法是提示用户关闭帐号保险箱 -\_-| 确实是很降低用户体验的一个不是办法的办法。**

**场景2：制作某游戏外挂，需要注入一个DLL到游戏进程中去直接调用游戏函数完成某一功能。结果发现该游戏有NP保护，OpenProcess打不开，创建远程线程也不行，试用其它方法也一一失败。遇到上面的情况，高手们自然是转到Ring0下面去，使用驱动之类的办法来对付啦，不过吾等菜鸟可就是酒井没法子了 -\_-|**

**不过也别太灰心，凡事总会有办法的。我想我们需要一种持久的、稳定的、不容易被安全软件屏蔽的DLL注入方法，后来发现，输入法程序就是能完成这一任务的理想人选。输入法程序程序到底是什么？它没有自己的进程，并且在系统还没有登录时就已被加载（在欢迎界面你也可以调出输入法），它可以在游戏中打开，也可以在控制台程序中打开，还可以在瑞星保护下的QQ中打开，在杀软中也可以打开，这不就是我们要找的特性吗。**

**那么，输入法到底是什么呢？根据Windows的规定，输入法其实就是一个DLL，不过它是一个特殊的DLL，它必须具有标准输入法程序所规定的那些接口，输入法是由输入法管理器（imm32.dll）控制的，输入法管理器又是由user32.dll控制的。输入法在系统目录是以IME为扩展名的文件，当在应用程序中激活某个输入法时，输入法管理器就会在那个应用程序的进程中加载对应的IME文件，注意，加载IME文件跟加载普通的DLL并没有本质区别，所以，可以认为，输入法其实就是注入到应用程序中的一个DLL文件，并且，这种“注入”是不会被杀软和游戏NP拦截的（至少目前是）。现在，我们已经有了一个注入DLL的另类方法，那就是利用输入法。**

**具体流程是这样，首先制作一个标准输入法文件，但是这个输入法并不完成文字输入工作，它的唯一任务就是用来注入DLL，所以称为“服务输入法”，然后，制作一个控制程序，来控制服务输入法，当然最后还需要一个用于注入的目标DLL，这样一共就有3个文件。开始工作后，控制程序首先将服务输入法安装到系统中，然后传递几个参数给服务输入法，参数中包括了需要注入的DLL文件的名称和路径，然后，控制程序将服务输入法设置为系统的默认输入法，这样新的程序一打开，服务输入法就会注入那个程序。**

**当然，在服务输入法安装之前打开的程序不会被注入，这时需要向系统中的所有窗口POST一条WM\_INPUTLANGCHANGEREQUEST消息，该消息可以在指定窗口中后台激活服务输入法，这样，系统中所有拥有窗口的进程就都被我们的服务输入法注入了。服务输入法注入程序之后，就会根据控制程序传递过来的参数加载目标DLL，这样目标DLL也就随着服务输入法一同注入到目标程序中了。注意服务输入法是控制程序用WM\_INPUTLANGCHANGEREQUEST消息在所有窗口中自动激活的，如果某个窗口自动激活失败，你就需要在那个窗口中手工切换到服务输入法，这样才能注入进去了。至于注入以后，你就可以在窗口中切换到别的输入法，这并不会影响已经注入进去的DLL。我将这一套功能制作成一个完整的示例，你可以在以下地址下载：** [**http://www.pen88.com/download/imehook.rar**](http://www.pen88.com/download/imehook.rar) **压缩包中的第6个和第8个文件夹演示了此功能并包含所有源代码。其中文件imedllhost09.dll就是服务输入法，运行时会被安装到系统中，控制程序退出时会自动卸载该输入法，这样用户就不太容易察觉，你还可以重新编译该输入法，将名称改为“中文（中国）”，这样隐蔽性更好。文件hxwdllwx.dll是演示用的目标DLL，你可以替换成自己的DLL，然后那个exe文件就是控制程序了。输入法imedllhost09.dll在运行时会被复制到系统目录并更名为imedllhost09.ime，它导出了2个函数用于控制。在VB中的声明为：**

**Public Declare Function IMESetPubString Lib "imedllhost09.ime" (ByVal RunDLLStr As String, ByVal UnloadDll As Long, ByVal loadNextIme As Long, ByVal DllData1 As Long, ByVal DllData2 As Long, ByVal DllData3 As Long) As Long**

**Public Declare Function IMEClearPubString Lib "imedllhost09.ime" () As Long**

**其中IMESetPubString用于向输入法传递要注入的DLL等参数。RunDLLStr，要注入的DLL命令和完整路径。UnloadDll，当输入法退出时，是否同时卸载目标DLL 0-是，1-否。loadNextIme，当切换至该服务输入法时，是否直接切换到下一个输入法（这样服务输入法就好像被跳过了，可最小限度影响用户的输入法顺序） 0-否，1-是。DllData1，DllData2，DllData3是传递给目标DLL的回调函数（函数名称必须为RunDllHostCallBack）的参数，你可以在目标DLL中导出一个函数，名称为RunDllHostCallBack，这样当输入法注入时会调用目标DLL的该回调函数并向其传递这3个参数。函数原型为（VC）：**

**DWORD RunDllHostCallBack(DWORD calldata1, DWORD calldata2,DWORD calldata3);**

**IMEClearPubString函数用于清除输入法的配置，清除后，输入法将停止在新的程序中注入目标DLL，但已注入的DLL不会卸载。**

**好了，利用输入法来注入DLL基本上就是这样了，详细的用法大家可以看压缩包中的第8个文件夹，其中服务输入法是VC写的，控制程序是VB的，代码都是有注释的。测试发现该方法能过目前所有杀软，也能注入冰刃。当然缺点还是有的，就是目标程序如果不接受输入法那就没办法了，但是现在一般的游戏都不会禁止玩家在里面打字吧，而且杀软也不能禁止用户输入汉字吧，哈哈，所以通用性应该还是蛮好的。**

**最后，我再介绍另一个注入DLL的方法，估计也很少被用到。是利用一个未公开函数RegisterUserApiHook，可以在网上搜索关键词“RegisterUserApiHook”，查到有人在Windows 2003下测试成功，但是我在Windows XP测试却失败。后来终于找到了失效的原因。RegisterUserApiHook函数可以在系统中注册一个全局钩子，你需要在钩子中指定一个DLL和一个回调函数，然后，所有加载了user32.dll的程序就都会在启动时加载你指定的这个DLL。用这个函数来注入DLL也是很不错的。但是测试发现它的注入能力似乎赶不上上面提到的利用输入法来注入的办法，可以注入一般的程序和某些安全程序，但是对冰刃无效。而且它有一个限制，就是系统中只能同时存在一个这样的钩子。实际上这个钩子平时是被系统中的Themes服务占用了，Themes服务正是利用这个钩子HOOK了绘制窗口的相关API，所以才让所有程序窗口变成XP主题样式的。所以我们要用这个钩子的话，必须先关闭Themes服务，这样在XP下也可以用了，但是这样系统就变成Windows 2000的样式了 -\_-|**

**RegisterUserApiHook函数的VB声明如下：**

**Public Declare Function RegisterUserApiHookXP Lib "user32" Alias "RegisterUserApiHook" (ByVal hInstance As Long, ByVal fnUserApis As Long) As Long**

**Public Declare Function RegisterUserApiHook2003 Lib "user32" Alias "RegisterUserApiHook" (pRegInfo As HookAPIRegInfo2003) As Long**

**可以看到，在XP和2003下这个函数的参数是不一样的。关于此函数的示例代码，请参见压缩包中的第5个文件夹。**

**最后的最后，再介绍一个未公开函数InitializeLpkHooks，这个函数在网上能找到的资料更少，只有一个声明而已。但是它名称中最后那个“Hooks”误导了我，我以为又是一个可以用来注入DLL的不错函数，用OD反出来一看，原来只是个局部HOOK而已。虽然没太大用，还是一并写上吧，也许谁用得着呢。InitializeLpkHooks顾名思义就是HOOK LPK的，Windows有个lpk.dll，就是支持多语言包的那么个功能。测试发现好多程序在TextOut之前似乎是要调用lpk.dll里面的相关函数的，可能是支持多语言的程序就需要用这个来判断到底要显示那种语言吧。而InitializeLpkHooks，就是用来HOOK lpk.dll里面的4个函数的，这4个函数是LpkTabbedTextOut，LpkPSMTextOut，LpkDrawTextEx，LpkEditControl。我们先打开VB，在窗体中加入以下代码吧：**

**Private Sub Form\_Load()**

**DLLhwnd = LoadLibrary("lpk.dll") '加载DLL**

**DLLFunDre = GetProcAddress(DLLhwnd, "LpkDrawTextEx") '获取回调函数地址**

**LpkHooksInfo.lpHookProc\_LpkTabbedTextOut = 0**

**LpkHooksInfo.lpHookProc\_LpkPSMTextOut = 0**

**LpkHooksInfo.lpHookProc\_LpkDrawTextEx = GetLocalProcAdress(AddressOf HookProc1) '设置要HOOK的LPK函数**

**LpkHooksInfo.lpHookProc\_LpkEditControl = 0**

**InitializeLpkHooks LpkHooksInfo**

**End Sub**

**Private Sub Form\_Unload(Cancel As Integer)**

**LpkHooksInfo.lpHookProc\_LpkTabbedTextOut = 0**

**LpkHooksInfo.lpHookProc\_LpkPSMTextOut = 0**

**LpkHooksInfo.lpHookProc\_LpkDrawTextEx = DLLFunDre**

**LpkHooksInfo.lpHookProc\_LpkEditControl = 0**

**InitializeLpkHooks LpkHooksInfo**

**FreeLibrary DLLhwnd**

**End Sub**

**然后新建一个模块，在模块中加入以下代码：**

**Public Declare Function LoadLibrary Lib "kernel32" Alias "LoadLibraryA" (ByVal lpLibFileName As String) As Long**

**Public Declare Function GetProcAddress Lib "kernel32" (ByVal hModule As Long, ByVal lpProcName As String) As Long**

**Public Declare Function FreeLibrary Lib "kernel32" (ByVal hLibModule As Long) As Long**

**' ----------------未公开函数--------------------------------------**

**Public Declare Sub InitializeLpkHooks Lib "user32" (lpProcType As Any)**

**Type LpkHooksSetting**

**lpHookProc\_LpkTabbedTextOut As Long**

**lpHookProc\_LpkPSMTextOut As Long**

**lpHookProc\_LpkDrawTextEx As Long**

**lpHookProc\_LpkEditControl As Long**

**End Type**

**' -------------------------------**

**Public DLLhwnd As Long, DLLFunDre As Long**

**Public LpkHooksInfo As LpkHooksSetting**

**Public Function GetLocalProcAdress(ByVal lpProc As Long) As Long**

**GetLocalProcAdress = lpProc**

**End Function**

**Function HookProc1(ByVal a1 As Long, ByVal a2 As Long, ByVal a3 As Long, ByVal a4 As Long, ByVal a5 As Long, ByVal a6 As Long, ByVal a7 As Long, ByVal a8 As Long, ByVal a9 As Long, ByVal a10 As Long) As Long**

**HookProc1 = 0**

**End Function**

**运行一下看看，是不是窗体中标题栏和按钮上的文字都没有了，因为我们把函数LpkDrawTextEx替换成自己的函数HookProc1了。这个函数有10个参数，其中几个好像是字符串指针，似乎可以用来截获窗体要显示的文字，然后改成另一种语言的文字，我猜想，也许就是这个用途吧。哈哈，纯属猜测。以上就是函数InitializeLpkHooks的用法了。**

**以上就是全部。**

**本文所有示例代码的下载地址是： http://www.pen88.com/download/imehook.rar**

**我的QQ511795070，欢迎交流。**