DLL注入技术之劫持进程创建注入

    劫持进程创建注入原理是利用Windows系统中CreateProcess()这个API创建一个进程，并将第6个参数设为CREATE\_SUSPENDED，进而创建一个挂起状态的进程，利用这个进程状态进行远程线程注入DLL，然后用ResumeThread()函数恢复进程。  
  
**1．创建挂起的进程**  
    下面是创建一个挂起的计算器程序进程的主要代码：

1. STARTUPINFO si = {0};
2. si.cb = sizeof si;
3. si.dwFlags = STARTF\_USESHOWWINDOW;
4. si.wShowWindow = SW\_SHOW;
5. PROCESS\_INFORMATION pi;
6. TCHAR cmdline[MAXBYTE] =\_T("calc.exe");
7. BOOL bRet = ::CreateProcess(
8. NULL,
9. cmdline,
10. NULL,
11. NULL,
12. FALSE,
13. CREATE\_SUSPENDED, //需要注意的参数
14. NULL,
15. NULL,
16. &si,
17. &pi);

复制代码

**2．向挂起的进程中进行远程线程注入DLL**  
    关于远程线程注入在这里就不重复讲述了，但是这里需要注意一个问题，那就是CreateRemoteThread()中第6个参数，需要设为CREATE\_SUSPENDED，主要参数如下：

1. //4. 创建远程线程
2. m\_hInjecthread = ::CreateRemoteThread(hProcess,      //远程进程句柄
3. NULL,                                            //安全属性
4. 0,                                               //栈大小
5. (LPTHREAD\_START\_ROUTINE)LoadLibrary,             //进程处理函数
6. pszDllName,                                      //传入参数
7. CREATE\_SUSPENDED,                                //默认创建后的状态
8. NULL);                                           //线程ID

复制代码

**3．激活进程中的线程**  
    这里主要用的是ResumeThread()的这个API，需要注意的是先激活主要线程，再激活注入的线程。  
  
    劫持进程创建注入其实就是远程线程注入的前期加强版，他可以在进程启动前进行注入，由于进程的线程没有启动，这样就可以躲过待注入进程的检测，提高的注入的成功率。