 Windows网络用户密码猜解算法的主要思想是:利用Windows提供的窗口枚举函数EnumWindows ()找到网络登录窗口。利用子窗口枚举函数EnumChildWindows ()或GetNext-DlgTabItem()和GetWindowLong()定位网络登录窗口上的各个控件。 利用SendDlgItemMessage() 或SetDlgItemText()来输入用户名及密码。利用SendMessage()发送“ 确定 ”消息。这样一来，就利用程序完成了整个网络登录过程。在重复这个过程中采用枚举的用户名和密码，进而完成网络用户名及密码的枚举猜解。 一、猜解过程流程：为说明问题，下面只写出主要的过程。对于关键过程给出用VC 实现的源码。下面的流程中Mutex.Lock和Mutex.UnLock之间的代码只允许单线程访问。“ 密码枚举完 ”是指用户指定的字符集合已被枚举完，程序将再枚举一个新的用户名，然后重新枚举这个字符集合。关于源码中各函数的具体用法，请参阅MSDN。关于多线程的用法，可参阅《VisualC 技术内幕》。 下面给出关键流程的源代码 1. 全局变量： struct _Thread { CWinThread *pThread. }. _Thread WindowThread[iProc],PassTread[1],UserTread[1]. )//iProc：窗口枚举线程数 CEvent gEventNextPass.//取下一个密码，为实现同步引进 CEvent gEventPassOk.//已取得密码，为实现同步引进 CEvent gEventNextUser.//取下一个用户名，为实现同步引进 CEvent gEventUserOk.// 已取得用户名，为实现同步引进 CMutex gMutex.//互斥量，只允许单线程访问 char

cCurrentPass[MAX_PASSWORD_LENGTH]. file://当前使用的密码。 char cCurrentUser[MAX_USER_LENGTH].//当前使用的用户名 2. 线程启动：{ file://密码枚举线程 if(PassTread[0].pThread==NULL) { PassTread[0].pThread=AfxBeginThread((AFX_THREADPROC)GetNextPassL,NULL, THREAD_PRIORITY_LOWEST). PassTread[0].pThread->m_bAutoDelete=TRUE. file://这里略去了从文件取得密码的代码，这些代码和用户名枚举过程的代码差不多 } file://用户名枚举线程 if(UserTread[0].pThread==NULL) { UserTread[0].pThread=AfxBeginThread((AFX_THREADPROC)GetNextUserF,NULL, THREAD_PRIORITY_LOWEST). PassTread[0].pThread->m_bAutoDelete=TRUE. } file://窗口枚举线程 for(int i=0.i { if(WindowThread[i].pThread==NULL){ WindowThread[i].pThread=AfxBeginThread((AFX_THREADPROC)ThreadProc,NULL, THREAD_PRIORITY_LOWEST). WindowThread[i].pThread->m_bAutoDelete=TRUE. } } 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com