1. 首先，一个进程要进行特权操作，例如修改其他进程内存数据的话，必须要有比较大的特权，一般来说，只有管理员账号有这个权限，那么我们程序的第一件事情，就是确认当前启动这个进程的账号，是否是管理员账号。

BOOL GetProcessElevation(TOKEN\_ELEVATION\_TYPE\* pElevationType, BOOL\* pIsAdmin) {

HANDLE hToken = NULL;

DWORD dwSize;

// Get current process token

if (!OpenProcessToken(GetCurrentProcess(), TOKEN\_QUERY, &hToken))

return(FALSE);

BOOL bResult = FALSE;

// Retrieve elevation type information

if (GetTokenInformation(hToken, TokenElevationType,

pElevationType, sizeof(TOKEN\_ELEVATION\_TYPE), &dwSize)) {

// Create the SID corresponding to the Administrators group

byte adminSID[SECURITY\_MAX\_SID\_SIZE];

dwSize = sizeof(adminSID);

CreateWellKnownSid(WinBuiltinAdministratorsSid, NULL, &adminSID,

&dwSize);

if (\*pElevationType == TokenElevationTypeLimited) {

// Get handle to linked token (will have one if we are lua)

HANDLE hUnfilteredToken = NULL;

GetTokenInformation(hToken, TokenLinkedToken, (VOID\*)

&hUnfilteredToken, sizeof(HANDLE), &dwSize);

// Check if this original token contains admin SID

if (CheckTokenMembership(hUnfilteredToken, &adminSID, pIsAdmin)) {

bResult = TRUE;

}

// Don't forget to close the unfiltered token

CloseHandle(hUnfilteredToken);

} else {

\*pIsAdmin = IsUserAnAdmin();

bResult = TRUE;

}

}

// Don't forget to close the process token

CloseHandle(hToken);

return(bResult);

}

1. 如果当前操作系统使用了UAT机制，那么我们当前运行的程序有可能是以过滤令牌的身份在运行，此时，如果我们想使用高级的权限，必须开通我们令牌中的一些特权，例如调试权限。这时，就用到了用户权限提升。

BOOL EnablePrivilege(PCTSTR szPrivilege, BOOL fEnable)

{

// Enabling the debug privilege allows the application to see

// information about service applications

BOOL fOk = FALSE; // Assume function fails

HANDLE hToken;

// Try to open this process's access token

if (OpenProcessToken(GetCurrentProcess(), TOKEN\_ADJUST\_PRIVILEGES,

&hToken)) {

// Attempt to modify the given privilege

TOKEN\_PRIVILEGES tp;

tp.PrivilegeCount = 1;

LookupPrivilegeValue(NULL, szPrivilege, &tp.Privileges[0].Luid);

tp.Privileges[0].Attributes = fEnable ? SE\_PRIVILEGE\_ENABLED : 0;

AdjustTokenPrivileges(hToken, FALSE, &tp, sizeof(tp), NULL, NULL);

fOk = (GetLastError() == ERROR\_SUCCESS);

// Don't forget to close the token handle

CloseHandle(hToken);

}

return(fOk);

}

1. 如果一个老的程序，没有考虑到特权问题，需要用户手动去提升权限，如何将这个老程序进行封装，使用户不用手动提升权限，直接运行这个程序。这个就用到了以管理员身份启动一个进程。

DWORD StartElevatedProcess(LPCTSTR szExecutable, LPCTSTR szCmdLine)

{

// Initialize the structure.

SHELLEXECUTEINFO sei = { sizeof(SHELLEXECUTEINFO) };

// Ask for privileges elevation.

sei.lpVerb = TEXT("runas");

// Pass the application to start with high privileges.

sei.lpFile = szExecutable;

// Pass the command line.

sei.lpParameters = szCmdLine;

// Don't forget this parameter otherwise the window will be hidden.

sei.nShow = SW\_SHOWNORMAL;

ShellExecuteEx(&sei);

return(GetLastError());

}

1. 如何让一个程序自动判断是64位操作系统，并选择合适的应用程序版本进行运行。

typedef BOOL (WINAPI \*LPFN\_ISWOW64PROCESS) (HANDLE, PBOOL);

LPFN\_ISWOW64PROCESS fnIsWow64Process;

BOOL IsWow64()

{

BOOL bIsWow64 = FALSE;

//IsWow64Process is not available on all supported versions of Windows.

//Use GetModuleHandle to get a handle to the DLL that contains the function

//and GetProcAddress to get a pointer to the function if available.

fnIsWow64Process = (LPFN\_ISWOW64PROCESS) GetProcAddress(

GetModuleHandle(TEXT("kernel32")),"IsWow64Process");

if(NULL != fnIsWow64Process)

{

if (!fnIsWow64Process(GetCurrentProcess(),&bIsWow64))

{

//handle error

}

}

return bIsWow64;

}

* 1. 先运行一个32为的程序，并判断当前程序是否运行在wow64下；
  2. 如果是，那么启动64为版本的应用程序，本程序推出
  3. 如果否，继续运行32位应用程序。