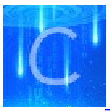
弹出UAC窗口让用户提权

华秋实 于 2015-03-30 20:26:58 发布 4521 收藏 1
分类专栏： C/C++ Windows

本文链接：https://blog.csdn.net/yockie/article/details/44756393
版权


C/C++ 同时被 2 个专栏收录
141 篇文章 7 订阅
订阅专栏

Windows
34 篇文章 0 订阅
订阅专栏

这里讨论的UAC 自我提权是指：先检查当前进程的等级，然后弹出窗口显示提醒用户进行授权。

最重要的代码：

```
            // 1.Check the current process's "run as administrator" status
            BOOL fIsRunAsAdmin;
            try
            {
                fIsRunAsAdmin = IsRunAsAdmin();
            }
            catch (DWORD dwError)
            {
                ReportError(L"IsRunAsAdmin", dwError);
                return;
            }


            // 2.Elevate the process if it is not run as administrator.
            if (!fIsRunAsAdmin)
            {
                wchar_t szPath[MAX_PATH];
                if (GetModuleFileName(NULL, szPath, ARRAYSIZE(szPath)))
                {
                    // Launch itself as administrator.
                    SHELLEXECUTEINFO sei = { sizeof(sei) };
                    sei.lpVerb = L"runas";
                    sei.lpFile = szPath;
                    sei.hwnd = hWnd;
                    sei.nShow = SW_NORMAL;


                    if (!ShellExecuteEx(&sei))
                    {
                        DWORD dwError = GetLastError();
                        if (dwError == ERROR_CANCELLED)
                        {
                            // The user refused the elevation.
                            // Do nothing ...
                        }
                        else
                        {
                            ReportError(L"ShellExecuteEx", dwError);
                        }
                    }
                    else
                    {
                        EndDialog(hWnd, TRUE);  // Quit itself
                    }
                }
            }
            else
            {
                MessageBox(hWnd, L"The process is running as administrator", L"UAC", MB_OK);
            }
        }
```

其中的IsRunAsAdmin用户判断当前进程是否以管理员权限运行，代码是：

```
//
//   FUNCTION: IsRunAsAdmin()
//
```

```
//
//   PURPOSE: The function checks whether the current process is run as
//   administrator. In other words, it dictates whether the primary access
//   token of the process belongs to user account that is a member of the
//   local Administrators group and it is elevated.
//
//   RETURN VALUE: Returns TRUE if the primary access token of the process
//   belongs to user account that is a member of the local Administrators
//   group and it is elevated. Returns FALSE if the token does not.
//
//   EXCEPTION: If this function fails, it throws a C++ DWORD exception which
//   contains the Win32 error code of the failure.
//
//   EXAMPLE CALL:
//     try
//     {
//         if (IsRunAsAdmin())
//             wprintf (L"Process is run as administrator\n");
//         else
//             wprintf (L"Process is not run as administrator\n");
//     }
//     catch (DWORD dwError)
//     {
//         wprintf(L"IsRunAsAdmin failed w/err %lu\n", dwError);
//     }
//
BOOL IsRunAsAdmin()
{
    BOOL fIsRunAsAdmin = FALSE;
    DWORD dwError = ERROR_SUCCESS;
    PSID pAdministratorsGroup = NULL;

    // Allocate and initialize a SID of the administrators group.
    SID_IDENTIFIER_AUTHORITY NtAuthority = SECURITY_NT_AUTHORITY;
    if (!AllocateAndInitializeSid(
        &NtAuthority,
        2,
        SECURITY_BUILTIN_DOMAIN_RID,
        DOMAIN_ALIAS_RID_ADMINS,
        0, 0, 0, 0, 0, 0,
        &pAdministratorsGroup))
    {
        dwError = GetLastError();
        goto Cleanup;
    }

    // Determine whether the SID of administrators group is enabled in
    // the primary access token of the process.
    if (!CheckTokenMembership(NULL, pAdministratorsGroup, &fIsRunAsAdmin))
    {
        dwError = GetLastError();
        goto Cleanup;
    }

Cleanup:
    // Centralized cleanup for all allocated resources.
    if (pAdministratorsGroup)
    {
        FreeSid(pAdministratorsGroup);
        pAdministratorsGroup = NULL;
    }

    // Throw the error if something failed in the function.
    if (ERROR_SUCCESS != dwError)
    {
        throw dwError;
    }

    return fIsRunAsAdmin;
}
```

完整代码在 这里

参考资料：

1.MSDN(https://code.msdn.microsoft.com/windowsapps/CppUACSelfElevation-5bfc52dd#content)