

Re: Issue with ASP.NET client, COM Interop, and Identity impersonation

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vc/2004-10/0063.html>

From: bruce barker (*nospam_brubar_at_safeco.com*)

Date: 10/01/04

Date: Fri, 1 Oct 2004 15:07:45 -0700

you need to set `aspcompat=true` (turn off thread agility) to lock down the thread calling the com component, then set the domain account and password in the web.config.

note: there is a small performance cost for `aspcompat=true`

-- bruce (sqlwork.com)

"Anil Krishnamurthy" <akrishnamurthy@nospam.air-worldwide.com> wrote in message news:OsByoV\$peHA.1296@TK2MSFTNGP12.phx.gbl...

> *We have an ASP.NET application that uses COM objects through Interop. The web application requires access to network and database resources and hence,*

> *needs to impersonate a domain account. The problem is that even when it is*

> *configured to run under a certain identity through Web.config, the*

> *impersonation is not carried through to COM library. Consequently, the*

code

> *in COM object runs under a local account and any code that needs to access*

> *network will not work correctly.*

>

> *ASP.NET*

> *{Web app} -----Interop ----->{COM Library}*

> *(Domain\NetworkUser)*

> *(LocalHost\IUSR_MachineName)*

>

> *We tried to solve the problem by impersonating the caller in COM library.*

> *Instead of using CoImpersonateClient(), which is required for every method,*

> *we tried using a different approach so that impersonation is in effect beyond function call. It was implemented as follows:*

>

> *HRESULT CClientImpersonator::Impersonate()*

> {

> *BOOL bOk = FALSE;*

> *HRESULT hr = E_FAIL;*

```
> DWORD dwErr = 0;
>
> // try to impersonate the client
> hr = ::CoImpersonateClient();
> if( SUCCEEDED( hr ) )
> {
> HANDLE hToken = NULL;
> HANDLE hDupToken = NULL;
>
> // get the thread's impersonation token
> bOk = ::OpenThreadToken( ::GetCurrentThread(), TOKEN_ALL_ACCESS,
> TRUE, &hToken );
> if( TRUE == bOk )
> {
> // dup it
> bOk = ::DuplicateTokenEx( hToken, TOKEN_ALL_ACCESS, NULL,
> SecurityImpersonation, TokenImpersonation, &hDupToken );
> if( TRUE == bOk )
> {
> // switch back the identity
> hr = ::CoRevertToSelf();
> if( SUCCEEDED(hr) )
> {
> // now impersonate the same identity so it 'sticks'
> beyond function call level
> bOk = ::ImpersonateLoggedOnUser( hDupToken );
> if( FALSE == bOk )
> {
> hr = HRESULT_FROM_WIN32( ::GetLastError() );
> ATLTRACE( "ImpersonateLoggedOnUser() failed.
> GetLastError() returned %8x.\n", hr );
> }
> }
> else
> {
> ATLTRACE( "CoRevertToSelf() failed. Error code
> %8x.\n",
> hr );
> }
> }
> else
> {
> hr = HRESULT_FROM_WIN32( ::GetLastError() );
> ATLTRACE( "DuplicateTokenEx() failed. GetLastError()
> returned %8x.\n", hr );
> }
> }
> else
> {
> hr = HRESULT_FROM_WIN32( ::GetLastError() );
> ATLTRACE( "OpenThreadToken() failed. GetLastError() returned
```

```
> %8x.\n", hr );
> }
>
> ::CloseHandle( hDupToken );
> ::CloseHandle( hToken );
> }
> else
> {
> if( hr != RPC_E_CALL_COMPLETE )
> ATLTRACE( "CoImpersonateClient() failed. Error code: %8x.\n",
> hr );
> else
> hr = S_FALSE; // ignore the failure when the context is not
> available
> }
> return hr;
> }
>
> Unfortunately, the solution does not seem to work on all machines. In
cases
> where it does not work, the behavior is as follows: Instantiation of COM
> object succeeds but attempt to invoke any method or access any property on
> the same object fails. The error is reported as
> "System.InvalidCastException. QueryInterface for interface <Interface
name>
> failed"
>
> Any ideas?
>
> Thanks
>
>
```