

Re: ASP.NET Impersonation / delegation

Source:

<http://www.tech-archive.net/Archive/Win2000/microsoft.public.win2000.developer/2004-04/0087.html>

From: Magdelin (*magdelinsuja_at_newsgroups.nospam*)

Date: 04/28/04

Date: Wed, 28 Apr 2004 14:41:08 -0700

Thanks Bruce. Are you sure, there will not be any security risk? The MS documentation does not recommend SE_TCB_NAME privilege to a any account other than the default LocalSystem.

I have included MS help description for the mentioned privilege

Act as part of the operating system

Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment

Description

This policy allows a process to authenticate as any user, and therefore gain access to the same resources as any user. Only low-level authentication services should require this privilege.

The potential access is not limited to what is associated with the user by default, because the calling process may request that arbitrary additional accesses be put in the access token. Of even more concern is that the calling process can build an anonymous token that can provide any and all accesses. Additionally, the anonymous token does not provide a primary identity for tracking events in the audit log.

Processes that require this privilege should use the LocalSystem account, which already includes this privilege, rather than using a separate user account with this privilege specially assigned.

By default, only the LocalSystem account has the privilege to act as part of the operating system.

Regards,
Magdelin

----- bruce barker wrote: -----

you are on the right track. ntlm will not delegate even if your security team allowed delegation, only digest allows delegation.

on win2k you have no choice but to set SE_TCB_NAME (this is removed in xp). its a fairly safe priviledge. its original intent was to control Trojan horses (programs that pretended to be the login program).

-- bruce (sqlwork.com)

"Magdelin" <magdelinsuja@newsgroups.nospam> wrote in message
news:9C2BA3A9-A627-46B0-9215-EA8C99E0F978@microsoft.com...

> *Hi all,*

>> *I am trying to implement ASP.NET impersonation/delegation in an intranet application in C#. The presentation layer developed in ASP.NET accesses the business logic via .net remoting. The business logic in-turn accesses the other network resources such as the SQL Server and the Active Directory.*

> *Both the business logic and the web application are deployed in IIS installed on two separate Win2k servers. Since, the application requires "Windows Authentication" in order to implement the declarative Role-based security, both business and presentation layers are configured for impersonation, by including the <identity impersonate="true"/> tag in their respective web.config files. The directory security of business and web applications hosted in IIS is configured for "Integrated Windows Authentication". The anonymous, digest and basic authentication options are not selected.*

>> *With the above mentioned configuration, if the business logic tries to access the active directory, a COMException occurs with the error message "An operation error has occurred". I believe this error has occurred because the impersonated account and the computer on which the business logic runs are not trusted for delegation by the Domain controller. The following links explains the reason for such an error.*

>> <http://support.microsoft.com/default.aspx?scid=kb;en-us:810572>

> <http://support.microsoft.com/default.aspx?kbid=325894>

> <http://support.microsoft.com/default.aspx?kbid=264921>

>> *Link to the newsgroup search*

>

<http://msdn.microsoft.com/newsgroups/managed/default.aspx?query=double+hop&dg=&cat=en-us-msdnman&lang=en&cr=US&pt=&catlist=&dglist=&ptlist=>

>> *Since our security team considers trusting win2k server for delegation to be a major security risk, I haven't had the opportunity, to test the business logic with the trusted configuration myself. From the trace log it is clear that the authentication type is NTLM and the account used to test the business logic has sufficient privileges to query the Active Directory (AD). The application is successful in querying the AD when account properties (userName and password) are included in the <Identity> tag.*

>> *Fortunately, I found few delegation alternatives in MSDN at*

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vsent7/html/vxconaspnetdelegation.asp>

>> *The WindowsImpersonationContext.Impersonate() is now considered as the best alternative for impersonating an account that is specially created for this purpose. The role based security will be implemented as before but for accessing resources such as AD and SQL Server the new helper account will be used. The account that currently runs the process will be impersonated with a special helper account which will have sufficient privileges to impersonate as well as query the AD. Once the task with the AD is completed, the windows identity will revert back to its original credentials. The following link details how to make such impersonation, possible.*

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/htm>

/cpconImpersonatingReverting.asp

- >> *When the impersonation and reversion is tried on the win2k server, I receive the error message "1314: The required privilege is not held by client". I know that the LogonUser API requires "Act as part of the operating system (SE_TCB_NAME)" privilege. But, I would like to grant the helper account with least possible privilege.*
- >> *Is there a privilege other than the "SE_TCB_NAME" that has fewer privileges but can still make the LogonUser API work? Is there a better alternative for ASP.NET impersonation/delegation?*
- >> *Any ideas or pointers to articles would be greatly appreciated.*
- >> *Thanks in advance.*
- > *Magdelin*