

Re: Server 2003 AD, security context APIs, "operations error" ??

Re: Server 2003 AD, security context APIs, "operations error" ??

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.platformsdk.security/2006-06/msg00130.html>

- *From:* "Joe Kaplan \ (MVP – ADSI)" <joseph.e.kaplan@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 30 Jun 2006 11:50:20 –0500
-

Hi Ollie,

Replies inline...

Note that if you aren't a .NET developer, you might not get much out of my book. However, we do include some interesting stuff regarding delegation and LDAP access (mostly with ADSI through .NET System.DirectoryServices) that does sort of apply to your scenario (although ours is couched in ASP.NET parlance, since that is a much more common server scenario than yours). If you do buy it, I hope you like it. :)

Joe K.

--

Joe Kaplan—MS MVP Directory Services Programming
Co—author of "The .NET Developer's Guide to Directory Services Programming"
<http://www.directoryprogramming.net>

--

"Ollie Jones" <olliejones@xxxxxxxx> wrote in message
<news:1151677304.735217.80280@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Thanks for the reply, Joe. (I will read your book — I have mountains of ignorance to cross). May I ask a couple of n00b questions as followup?

Joe Kaplan (MVP – ADSI) wrote:

2003 AD doesn't support anonymous searches by default, so that's probably the difference.

This could indeed be the problem. Are you saying that the NTLM security context retrieved from the client isn't passed along to the AD? Is the act of passing along the security context the thing called "delegation" in Windows security parlance?

Re: Server 2003 AD, security context APIs, "operations error" ??

Yes, that's exactly it. A Windows service cannot forward a user's security context to a remote service automatically. It depends on a feature of Kerberos called delegation to be able to do that.

My guess is that your code has never been successfully authenticating with AD and has been binding as the anonymous user all along.

This is probably right. The server AD get call fails another way -- actually it fails trying to read the server machine's local registry according to sysinternal's regmon -- if the server process doesn't have local machine administrator privileges.

I could see this happening as well, although this is more likely just a permissions issue on the registry key in question. What did regmon say about the user who was denied access? If it was the impersonated user, then your impersonation is fine, but that user doesn't have the right permission on the key anyway. If it said the user was anonymous, then you somehow impersonated the anonymous user.

Since you have a multiple machine hop in here but can't delegate, you'll end up with anonymous authentication to AD via NTLM.

So the get function to the AD first tries using the thread's security context and if that fails it tries anonymous access, is that correct?

It is sort of like that. The function actually tries to use the current security context to log on remotely, but if it is not forwardable (cannot be delegated), then it just chooses to authenticate remotely as the anonymous user. This won't fail from an authentication perspective, but what often happens is that AD then automatically fails the request because it doesn't allow the anonymous user to perform operations.

Is it important to use the client's security context to hit AD via LDAP?

What I have to do is retrieve an ExtendedRight that is set for the client on a serviceConnectionPoint underneath the server's computer directory entry. If the client doesn't have that right set, the server is to refuse the client's request. The server certainly don't

Re: Server 2003 AD, security context APIs, "operations error" ??

Re: Server 2003 AD, security context APIs, "operations error" ??

need the client security context to carry out the operation once it's authorized. What I am not clear on is whether the server must impersonate the client to perform the check. Obviously everything would be simpler if it didn't.

It sounds to me that you could easily use the server's process identity to access AD and read this info if the process' account has the rights in AD to read the data in question. Based on what you said, it sounds like it does. The basic thing you would do is either not impersonate in the first place or call `RevertToSelf` before doing ADSI. To ensure that your server has the right permissions, you need to ensure that the account it runs under has the proper permissions in AD to read the data in question and that the service runs as a domain account so that it can authenticate with AD in the first place. System and network service will both use the machine account's credentials when accessing the network, so if you are using either of those as your service account and the machine is a domain member, that should work. You could also configure the process to run under a specific domain service account. It won't work at all if the machine is a workgroup computer though. In that case, you would need to call `ADsOpenObject` with hard-coded domain credentials that you would then need to store in a secure way.

Any wisdom?
Thanks.