

Re: Trusted SQL Connections & NT AUTHORITY\NETWORK SERVICE

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2007-03/msg00004.html>

- *From:* "Joe Kaplan" <joseph.e.kaplan@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 1 Mar 2007 14:13:08 -0600
-

Your summary is correct. BTW, the same rules apply when using the local SYSTEM account in terms of the credentials it uses on the network. The great advantage of Network Service is that to the local machine, it has very few privileges and can do much less damage locally on the server if compromised. That's why it was invented. Local Service is like Network Service, but it doesn't have any network credentials at all, so you can't use it in services that access the network in any reasonable way.

When you implement a trusted subsystem security architecture, which is what you are doing when you use a fixed account to access your backend data resources, there are many debates over whether to use a fixed service account vs Network Service. Generally, I think it is best to use Network Service when possible as it gives you one less password to manage, it has a lot of useful ACLs and group memberships out of the box as well as a bunch of useful local security policy privilege assignments such as "run as a service" and "generate security audits". Also, the machine account will generally (but not always) get the correct Kerberos servicePrincipalNames set for various services that are installed on the machine, which allows you to do Kerberos auth more easily. This is a good thing.

You can do all of those things manually with a fixed service account, but there is more effort.

In some cases, you really do need to use a fixed service account to do something you are doing with load balancing or something where you need two instances of a service on two different boxes to have the same Kerberos SPNs.

You can solve the SQL permissioning issue either way by creating an additional abstraction of a group to delegate permissions to. Then you can add whatever accounts you need to that group and it ends up not matter much to SQL.

So, in summary, it depends. :) It is good to know your options and what the affects might be of pulling one lever vs. another.

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>

"Craig Wagner" <MSDNNospam207@xxxxxxxxxxxxxxx> wrote in message
<news:D674D264-AA24-49E2-A530-46C5D7BD6ED3@xxxxxxxxxxxxxxx>

It was as you suspected. I wasn't seeing the forest for the trees. I could have sworn I double-checked all the settings in every location, but I missed the web.config where the database connection string was. At home it was hitting a SQL Server on the same machine as the web app. I changed it to my other database server and then I saw the domain\machine\$ being used to connect.

So to summarize and confirm what I've discovered...

If you have an application configured to run as Network Service:

- domain\machine\$ is the credentials that are exposed outside of the machine on which the app is running
- NT AUTHORITY\NETWORK SERVICE is the credentials exposed when accessing resources on the same machine on which the app is running
- when setting up the SQL login (using SQL Management Studio), I cannot browse for domain\machine\$ in Active Directory, I must know the name and manually type it into the New Login dialog

One more question regarding Best Practices.

If I'm setting up my app on a web farm I have two choices:

1. Run the app as Network Service on each web server and grant each web server's machine account access to the SQL Server.
2. Create a custom domain account and run the app as that identity on each web server. Then I only have to create a single login on the SQL Server.

From an on-going operations perspective, it seems like option (2) is the better way to go, as I don't have to add/remove logins from the SQL Server when I add/remove machines from the web farm. But I don't know what problems

I may create or conveniences I may lose by going that route.

As always, if you know of a source just point me in the right direction. I've not been able to find definitive, plain English answers to these

things.

"Joe Kaplan" wrote:

Are you sure the web app is trying to hit SQL on the network? The network service account is supposed to use the credentials of the machine account when it accesses the network to use a remote resource. What you are seeing at work is what I'd expect and what you are seeing at home is a bit strange to me. Any other environmental differences?