Re: Server Error in '/MyWebForm' Application

Source:
http://www.tech−archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2004−09/3772.html

---

From: Greg Burns (*greg_burns_at_DONT_SPAM_ME_hotmail.com*)
Date: 09/17/04

`Date: Thu, 16 Sep 2004 23:47:10 −0400`

comments inline...

"Karl" <karl REMOVE @ REMOVE openmymind REMOVEMETOO . ANDME net> wrote in message news:O28PbRGnEHA.3712@TK2MSFTNGP15.phx.gbl...
> Greg, I'm very tired so I'm not 100%...I meant Integrated Security in the
> connection string. I would say the vaste majority of people are using an
> SQL account and not Integrated Security.

I am not sure what most do, but I've always read to use Integrated Security w/ the ASPNET account

> I would also say the vaste majority of people don't use impersonation.
> Impersonation is only really
> handy in intranets where users are all on the same domain....or workgroup
> or
> whatever they call then (not a network guy).

Yes, I agree with that. Intranet is what we call them in these parts. :)

>
> Back to the connection string. I realize that SSPI is what microsoft
> recommends but I just don't think that's what most people are using. This
> might make no sense, but isn't the ASPNET account local to the webserver?
> Would it be able to access an SQL database on a different server even if
> they were on the same domain? other than that (which is obviously a deal
> breaker), not sure why I'm so anti−integrated security....

I had this discussesion a few weeks ago with somebody else here who thought the same.

The ASPNET user is a local account on the webserver. You are correct that this will be an issue when trying to connect as that user to SQL on another server in same domain.

Two solutions:
#1 Change machine.config's <processModel> so that it doesn't use local ASPNET user, but instead uses a domain account. If you do this, you must

give this domain user the equivalent rights as the ASPNET user. (Read PITA)

#2 Change machine.config's <processModel>, but this time keep ASPNET user (ie, user="machine") but change password from "autogenerate" to a known password. Reset ASPNET user's password to that known password.
Now on your SQL server, create a local account named (wait for it...) ASPNET. Set the password to the same as on web server. Believe or not, this WILL work. It is referred to as Pass−through authentication. Now there is some issue with the "number of hops", but there my expertise ends, cause I also am not a network guy. :^)

In fact, method #2 is recommended by MS. Check out: "Building Secure ASP.NET Applications" chp. 12 p299
or on the web

http://msdn.microsoft.com/library/default.asp?url=/library/en−us/dnnetsec/html/SecNetch12.asp

under: Using the ASP.NET process identity

It is late here also, time to call it quits.

g'night!
Greg