

Re: DCOM calls fails – access denied

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2002-11/1882.html>

From: Stefu (stefanroth@hotmail.com)

Date: 11/15/02

From: "Stefu" <stefanroth@hotmail.com>

Date: Fri, 15 Nov 2002 10:35:10 +0100

Hi iulian

Thanks for the answer! That's exactly how I understood the ASP.NET security. But why does one configuration work but not the other? The only difference is the processModel tag in machine.config. In both cases the worker thread should get the token from IIS (configured domain user). That's why I'm confused. In theory both versions should work?!

Thanks Steve

"iuga" <iulian.iuga@audicon.net> wrote in message
news:#kQkX6#iCHA.1960@tkmsftngp08...

> Hi,

>

> It's very important to understand how ASP.NET security works together with
> IIS security. For that I recommend you to read this article from msdn:

>

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

> [authaspdotnet.asp](#) .

>

> Now regarding your security settings:

>

> 1) In both cases you have in the web.config file the identity tag
> present with the impersonate attribute set to true. That means the worker
> thread will use the token it gets from IIS. If you would have the
> impersonate attribute set to false or the identity tag would be not
present

> in your web.config file, then the worker thread would get the token from
the

> worker process (aspnet_wp.exe). In machine.config file you can set, in
> context of which user you want to run the worker process using
processModel

> tag. As you can see, in your case, doesn't matter what you set in

> machine.config it doesn't affect the security context in which your worker

> thread is running.

>
> 2) IIS security configuration. You have set Anonymous access to the
web
> application. That means the IIS will send to the worker thread the token
of
> the user you have set for anonymous access. (Note: Take care about other
> settings there!!! If you have also Integrated Windows authentication
> checked, then the IIS will get information about LAN user accessing the
> application and it will send to the worker thread the token of this user
and
> not anonymous.) Ok, so far you can determine in context of which user the
> worker thread is running, now we have to see the remote access for this
> user.
>
> 3) DCOM application. You wrote in both configurations that the access
> and launch permissions are restricted, but without details which are the
> restrictions . Here you have two situations, depending the user you set as
> anonymous for IIS. If you set there a domain account, then you have to be
> sure that this account has at least access permissions (and launch
> permissions if you want to allow it to start the application) for DCOM
> application. If you set the IIS anonymous user a local user from the
server
> where IIS is running then again you can have two possibilities, you have a
> user on the computer where DCOM is running with the same name, then you
> should have similar passwords for this user on both computers and be sure
> this user has at least access permissions to DCOM application. Or the
second
> possibility, you don't have the same user to both computers, then the user
> is resolved as guest (anonymous logon) and be sure you don't have the
Guest
> account disabled and it is allowed to access the DCOM application.
>
> I hope you will have a better understanding about what is happening there
> and then you can adapt the settings to have the system running.
>
> Iulian Iuga,
> <http://www7.brinkster.com/iiuga>
>
>
>
>
>
>
>
> "Stefu" <stefanroth@hotmail.com> wrote in message
> news:evhi7A#iCHA.2256@tkmsftngp12...
> > Hi
> >
> > I'm a little bit confused by the security features of ASP.NET. In my web
> > application I have to call several DCOM Servers located on a remote
> > server.
> > Since not all users have a domain account, I had the idea to allow only

microsoft.public.dotnet.framework.aspnet.security: Re: DCOM calls fails – access denied

> > *anonymous access on my site. The users will be authenticated by a custom
> > form and a database. In order to have access rights to call the DCOM
> servers
> > I changed the default account for the web site to a domain user. In the
> > web.config file I changed the identity tag to <identity
impersonate="true"
> > />. As I saw in a microsoft .NET Security Guidance, this should lead to
a
> > impersonated call (identity = configured domain user). But all calls
> fail –
> > access denied. Only if the processModel tag in machine.config is changed
> to
> > userName="domain\user" password="password" the call works. But this
isn't
> > exactly the configuration I'm looking for. I do not want to run the
worker
> > process under a domain account with extended privileges. Is this
possible?
> > What else is necessary?
> >
> > Summary
> >
> > The following configuration is WORKING
> >
> > ASP.NET
> > web.config
> > ...
> > <identity impersonate="true" />
> > ...
> >
> > machine.config
> > ...
> > <processModel ... userName="domain\username" password="password" ...
> />
> >
> > IIS
> > only anonymous access
> > default account for my site changed to domain\username
> >
> > DCOM
> > Authentication Level = None
> > Identity = specific account
> > Security = access and launch permissions restricted
> >
> > -----
> >
> > This configuration does NOT WORK
> >
> > ASP.NET
> > web.config
> > ...*

Re: DCOM calls fails – access denied

microsoft.public.dotnet.framework.aspnet.security: Re: DCOM calls fails – access denied

```
> > <identity impersonate="true" />
> > ...
> >
> > machine.config
> > ...
> > <processModel ... userName="SYSTEM" password="AutoGenerate" ... />
> > <----- different
> >
> > IIS
> > only anonymous access
> > default account for my site changed to domain\username
> >
> > DCOM
> > Authentication Level = None
> > Identity = specific account
> > Security = access and launch permissions restricted
> >
> > Thanks in advance Steve
> >
> >
>
>
```