

Authentication Sharing Across Apps

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2006-10/msg00031.html>

- *From:* PolarBears <PolarBears@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 29 Sep 2006 13:23:01 -0700
-

Ok. For my part "B" question that I had (Login App was not returning authentication to calling app), I found the solution. I did not realize there were additional requirements for having applications communicate in .NET 2.0. Basically, in both the Login App and Calling App Web.Config, I did the following:

```
<authentication mode="Forms">  
  
<forms name=".ASPXAUTH" loginUrl="/Login.aspx"  
protection="All" enableCrossAppRedirects="true" path="/">  
  
</authentication>
```

Notice the addition of `enableCrossAppRedirects="true"`. That was different from .NET 1.1. Name, Path, and Protection must be identical on both the Login and Calling Apps.

Also, the `validationKey` and `DecryptionKey` must be identical on both apps. By default, they are autogenerated and different among apps, so you have to manually generate them, for example put something like this in the Web Config:

```
<machineKey  
validationKey="32F090935F6E49C2C797F69BBAAD9702ABD2FD0B667A8B44EA7DD4374267A75D7AD97746  
  
decryptionKey="CBAA84D7EC4BB56D75D217CECFB9629908CEB8BF91CFCD64568A145BE59719F"  
  
validation="SHA1"  
  
decryption="AES"  
  
>
```

I hope this helps somebody out there. I know sometimes finding the answer you are looking for can be real challenging if you don't ask the question a certain way.

Authentication Sharing Across Apps

Thanks,

Danny

"PolarBears" wrote:

Ok. Making progress. I had Windows Authentication and when I changed to SQL Integrated, the web service worked. On the Web Page (my original problem) I impersonated an account to get that working. Almost there!!! Just one little glitch and I'm home free.

The situation now is. From a single web app, I have a login screen and everything works fine, I get a message back "user logged in", that I had set up. Great.

However. When I call the login page from another .NET application, the login page does not return to the other application. The query string is up in the address pane, ready to go, but it just clears the login boxes and sits there. No error. I'm using the same Membership database and parameters on both apps, both are .NET 2.0. It does everything but go back like it should.

The point being we want to create a single login application and have our other applications all authenticate off of this application. We had it working in 1.1, but I'm having a hangup here. Ideas?

Thanks!

Danny

"Joe Kaplan" wrote:

Unfortunately, your analysis here is most likely wrong. There is nothing really different between how .NET 1.1 and 2.0 attempt to establish an authenticated connection with SQL server. If you use a connection string that uses Windows authentication in both cases, then the authenticated connection will attempt to be established based on the security token that is being used to execute the current thread at the time the code is called. That will either be the security token of the process, or the security token of some impersonated identity.

Whether or not the remote service accepts the authentication from client (the web server in this case) depends on whether there is a trust relationship between the account being authenticated and the remote service and whether or not the identity being authenticated is being delegated by the intermediate service (and whether delegation is allowed and configured).

Authentication Sharing Across Apps

You can discover whether you are getting an apples to apples comparison here

by determining first whether the current security context in both web applications is the same account. Check `System.Security.Principal.WindowsIdentity.GetCurrent().Name`. If they are both the same and one can authenticate to the remote SQL server but the other can't, it is almost certainly because one web application can do Kerberos delegation and the other one can't. If they aren't the same account (one is the authenticated user and one is the web server process account), then you aren't comparing the same thing.

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>

--

"PolarBears" <PolarBears@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message

news:C6AEC9E4-B964-474D-8D35-3B1E58BEFE06@xxxxxxxxxxxxxxxxxxxx

I've been away from this issue for a while, with other things. Here's an interesting twist that may shed some light. Again, let me set it up. SQL Server 2000 is on one machine with the database in question. Web Server is on a second machine.

Approaching from a different direction, I tried this. Using Visual Studio Tools for Office 2005, I created two buttons and two datagrids. The first datagrid accesses file "X" via a .NET 1.1 Web Service. The second datagrid accesses the same file "X" via a .NET 2.0 Web Service. The web services are on the same machine.

Visual Studio Tools for Office 2005 is running on .NET 2.0. Now, here's the interesting part, when I click the button to load the data from the .NET 1.1

Authentication Sharing Across Apps

Web Service, everything loads fine. Duplicated the same process with the .NET 2.0 Web Service, Login failed for User "Null". This tells me this is probably something specific to .NET 2.0 and not SQL Server or the Web Server, else it wouldn't work with .NET 1.1.

-- Danny