

# WindowsTokenRoleProvider Anyone?

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2006-03/msg03915.html>

---

- *From:* progrock <"prog\_rock{at}msn[dot]com">
  - *Date:* Thu, 23 Mar 2006 13:41:41 -0800
- 

Hey All, I'm attempting to put together a "secure" ASP.NET 2.0 application with one requirement that has given me a bit of grief: In a nutshell, if the user's session expires then they should be forced to re-authenticate with the application by providing logon credentials. These credentials are currently Active Directory domain accounts. This is to prevent the user from walking away from their workstation and another user walking up and accessing the application using their perhaps still unexpired session. Yes, I know there are better ways to enforce this kind of security but lets pretend the web app has to do it all, 'kay?

In my experience, the easy way to implement security with domain users is to use the Windows Authentication model built in to ASP.NET. The problem with this is that the browser, IE6, caches any previously supplied credentials until it is closed. So, once they log in to the app the first time they never get prompted to do it again...even if their session expires. Only way to clear the credential cache is to close the browser. That won't work as it needs to be implicit. They just carelessly walked away, remember?

So, to have the programmatic control over the authentication mechanism seems to leave only one choice in this scenario: Forms Authentication. Hmm, how to get Forms Auth to secure an app so that only domain users in a given global group are permitted to log in? 2.0 Membership provider to the rescue! Well, not exactly. "ActiveDirectoryMembershipProvider" and "WindowsTokenRoleProvider" seem to be up to the task, but I'm hitting an error not even the mighty Google Search can shed any light on:

[Begin Error]

Method is only supported if the user name parameter matches the user name in the current Windows Identity.  
Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Configuration.Provider.ProviderException: Method is only supported if the user name parameter matches the user name in the current Windows Identity.

[Clip]

Stack Trace:

[ProviderException: Method is only supported if the user name parameter matches the user name in the current Windows Identity.]

System.Web.Security.WindowsTokenRoleProvider.GetCurrentWindowsIdentityAndCheckName(String userName) +2182633

[End Error]

## WindowsTokenRoleProvider Anyone?

I don't know what to make of this. The error seems to imply the need for impersonation, which would be bad if true. Problem is, even with impersonation turned on there is still no joy (same error). Another thought is that it wants the full domain\account syntax in the login control. No, that just fails to authenticate entirely. It's probably something obvious and I'm just too tired to think straight.

My implementation is straight out of MSDN AFAIKT. So, before I go down the custom-provider road which overrides the offending method into oblivion, does anybody have any other ideas? Thanks for reading this far!

-Lee

Here is my web.config for those patient enough to endure all this:

```
<!-- ***BEGIN SECURITY CONFIGURATION*** -->
<authentication mode="Forms">
<forms name=".AMSFORMSAUTH"
loginUrl="~/Logon.aspx"
defaultUrl="~/Default.aspx"
protection="All"
timeout="10"
path="/"
requireSSL="true"
slidingExpiration="true"
cookieless="UseDeviceProfile"
domain=""
enableCrossAppRedirects="false" />
</authentication>
<authorization>
<deny users="?" />
<allow roles="[The domain group for this app]" />
<deny users="*" />
</authorization>
<membership defaultProvider="ExtranetActiveDirectoryMembershipProvider">
<providers>
<add name="ExtranetActiveDirectoryMembershipProvider"
connectionStringName="ActiveDirectoryConnectionString"
connectionUsername="[Removed domain account]"
connectionPassword="[Removed account password]"
attributeMapUsername="sAMAccountName"
type="System.Web.Security.ActiveDirectoryMembershipProvider, System.Web, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
</providers>
</membership>
<roleManager defaultProvider="ExtranetActiveDirectoryRoleProvider"
enabled="true"
cacheRolesInCookie="false">
<providers>
<add name="ExtranetActiveDirectoryRoleProvider"
type="System.Web.Security.WindowsTokenRoleProvider" />
</providers>
</roleManager>
```

## WindowsTokenRoleProvider Anyone?

```
<!-- ***END SECURITY CONFIGURATION*** -->
```

The connection string:

```
<add name="ActiveDirectoryConnectionString" connectionString="[Removed perfectly good LDAP path to domain]" />
```

.