

Re: Query AD using Integrated Authentication?

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2004-07/0323.html>

From: Joe Kaplan \((MVP - ADSI)\) (joseph.e.kaplan_at_remove_this.accenture.com)

Date: 07/27/04

Date: Tue, 27 Jul 2004 15:24:11 -0500

You don't have a password with integrated auth, so essentially, you are trying to do a bind with a username and a blank password. That won't work for sure and if you try it very often, you'll lock out that poor user.

The way you have to do this with WIA is to impersonate the logged on user (via your web.config) and don't supply any credentials. Then, ADSI will use the credentials of the current security context (the user you are impersonating) to contact AD.

The trick here is that if the AD server is on a different machine (very likely), you'll need to Kerberos Delegation to get this impersonation to work. Read these articles:

<http://support.microsoft.com/default.aspx?scid=kb:en-us:329986>

<http://support.microsoft.com/default.aspx?scid=kb:en-us:810572>

Good luck,

Joe K.

"Dave" <Dave@discussions.microsoft.com> wrote in message news:0FAA654B-B390-416D-99F8-18F0E39D226C@microsoft.com...

> Hi,

>

> *I want to query AD for user's information once they are logged in.*

>

> *Under Basic authentication, this worked fine using the code below.*

>

> *However, when I switched to Integrated for an intranet site, the FindOne()*

bombs with "Logon failure: unknown user name or bad password". I don't know how to pass the username/password information while using Integrated Security. Is there a way to do this?

>

> *System.DirectoryServices.DirectoryEntry entry = new*

System.DirectoryServices.DirectoryEntry("GC://mycompanydomain.com",

HttpContext.Current.Request.ServerVariables["AUTH_USER"],

HttpContext.Current.Request.ServerVariables["AUTH_PASSWORD"]);

> *System.DirectoryServices.DirectorySearcher search = new*

microsoft.public.dotnet.framework.aspnet.security: Re: Query AD using Integrated Authentication?

```
System.DirectoryServices.DirectorySearcher(entry);  
>  
> search.Filter = "(sAMAccountName=" + sSAMAccountName + ")";  
> search.PropertiesToLoad.Add("sAMAccountName");  
> search.PropertiesToLoad.Add("cn");  
> search.PropertiesToLoad.Add("givenName");  
> search.PropertiesToLoad.Add("sn");  
> search.PropertiesToLoad.Add("mail");  
> search.PropertiesToLoad.Add("telephoneNumber");  
>  
> System.DirectoryServices.SearchResult result = search.FindOne();  
>  
>
```