

Re: Windows Authentication, Single sign on and Active Directory

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2007-02/msg00045.html>

- *From:* "Joe Kaplan" <joseph.e.kaplan@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 12 Feb 2007 12:10:33 -0600
-

The best thing to do would be to catch the appropriate exception if the web service proxy client fails to connect due to authentication failure and then prompt the user for credentials. You can then create a NetworkCredentials object that contains the plain text credentials and use that instead of the DefaultCredentials. This gives you SSO for users logged in to the desktop but gives you a mechanism to handle situations where it doesn't work. This is actually how the browser tends to work as well.

I can't remember the exact exception, but I'm sure you can figure it out quickly through a little testing.

If you were working in an internet scenario instead of intranet, you might also want to consider the possibility of using Basic authentication with SSL. That will require you to always prompt for credentials, but is the most flexible. Note that you should almost certainly be using SSL with the web services anyway, as it is generally important to protect any authenticated web traffic at the transport level, no matter what authentication protocol you are using.

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>

"SP" <spsp@xxxxxxxxxxxx> wrote in message
news:OhJUeZsTHHA.3440@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Thanks for the input Joe,

I do not want to implement my own protocol and anything readymade is definitely the best option for me. I just got carried away reading lots of different articles on internet but could not get anywhere in practice. I think I was looking to replace the existing mechanism on a like for like substitute.

Re: Windows Authentication, Single sign on and Active Directory

One of the problems is, I need to keep the login dialog as well, in case if a user logs in to the client machine outside the domain, then he should be able to key in his credentials. The server is always in the domain. for example, I am a domain user as MyDomain\SP. On a client machine, if I am logged in as one, then I want the application to not show me a login dialog. If I am not logged on to the domain, I would like for the application to show me a login dialog where I may enter MyDomain\SP as user and my password to start using the application

As I had previously said, I don't know how to do it so can you please point me to some examples if possible?

TIA,

--SP

"Joe Kaplan" <joseph.e.kaplan@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:%23CmLAXsTHHA.3996@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

The more straightforward way to do this is to enable integrated Windows authentication on the web services. If the user is logged into the domain when running the client app, all you need to do is set the Credentials property on your web service proxy classes to use DefaultCredentials and the user will log in automatically (assuming the web server is also a member of the domain).

Your web services can then determine the identity of the authenticated user with Context.User.Identity.Name.

The other alternative to consider would be to use WCF or WSE3 or something to implement some sort of message level security.

I would recommend that you NOT try to implement your own authentication protocol. It is not easy to get right.

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>

--

"SP" <spspsp@xxxxxxxxxxxx> wrote in message news:elcz2irTHHA.496@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Hello All,

First of all, let me make it very clear that I do not have any idea about implementing the windows authentication, so all inputs

Re: Windows Authentication, Single sign on and Active Directory

would be appreciated.

The scenario :

I have a client – server application. At the moment, the passwords for the users are stored in a password storage (encrypted). The client app shows a login dialog, gets the username and password and sends it to the server where the server verifies it against the password store. I would like to move to a position where the user does not need to enter the username and password, If they are logged on to the domain, they should go straight in. The application is written in C# (client app) accessing the ASP.NET web services. All of it is done in .NET framework 2.0

The way I have figured out so far is as follows :

On the client side,

- 1) Get the user's identity
- 2) Send this to the server

On the server side :

- 3) Validate the user's identity against the active directory
- 4) If the user is valid, the normal process of using the application continues.

In the process of trying this, I have done the following:

- 1) Get the user's identity

```
System.Security.Principal.WindowsIdentity.GetCurrent().User.Value
```

- 2) send it to server (the value returned from the above call is string)

so I send it as it is.

On the server side

- 3) I try and create the SecurityIdentifier object as follows :

```
System.Security.Principal.SecurityIdentifier sid = new SecurityIdentifier(sddlIdentity);
```

this call is okay. From here I don't know where to go and how to

validate this against the active directory. I had a look at AD objects

and it seems the SID should be available in some tokengroups but this is

where I have got completely lost. (On a sidenote :Another thing is, If I

try and use AD searching, I get an error possibly because the

Re: Windows Authentication, Single sign on and Active Directory

call is
run as ASPNET user which does not have access to AD)

Kindly help me in achieving this or if this method is not the
correct
way of achieving my goal, advise me accordingly.

TIA,

SP