

Re: ADFS Development Issues

Source:

http://www.tech-archive.net/Archive/Windows/microsoft.public.windows.server.active_directory/2007-04/msg01138

- *From:* CJ <conorjgallagher@xxxxxxxxx>
 - *Date:* 19 Apr 2007 20:46:53 -0700
-

Hi Joe,

I had predicted you would be the first to reply to my post! You seem to be a resident ADFS expert in the newsgroups. Thanks for getting back to me on this.

I thought that what I was trying to do wasn't really possible. As you said I could hack the logon process, but I completely agree that this should be avoided if possible. So I think I may have to go back to the drawing board and try to directly interrogate the AD itself using System.DirectoryServices. Not quite sure how this is going to work yet, will have to have a look at the functionality this provides, have a think about it and how it fits into our project structure. Any pointers on this would be appreciated!

This does lead me to a further question though if you don't mind helping me a little more? We have a situation where we spawn another third party website in a new browser (not the same third party web app that passes us the logon details as per original post). We need this site to be automatically authenticated by our windows application so that the user will not be prompted for the details. This site will be using ADFS (claims aware style) so I am wondering how do we get the windows application to do this? We will obviously have the username and password from our logon screen so will just need a way to stop the automatic popup of the logon form. Is there a way we can create these credentials and push them through so that we can auto logon? (Note: This will be a pop up browser so it will be IE handling it and not a built in webbrowser control)

Can you give me any advice on this?

Thank you for your help!
Conor.

On Apr 20, 2:43 am, "Joe Kaplan"
<joseph.e.kap...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

You are asking for trouble with this type of design becauseADFSV1 is only

Re: ADFS Development Issues

intended to be used with browser clients, not programmatic "active" clients like a web service proxy. The issue is that the WS-Federation Passive Requester Profile (PRP), which is what ADFS V1 implements, uses a protocol based on redirects and possibly uses forms-based authentication to collect credentials. Typically, web service proxies don't handle this type of thing well. The redirects might work, but forms auth is proprietary. Also, the proxy client might get redirected to the home realm discovery page, which is not really something the proxy client can deal with.

A future version of ADFS, V2 most likely, will support the WS-Federation Active Requester Profile (ARP). ARP has first class support for web services and will integrate with .NET 3.0/WCF using the federation profile. However, that isn't shipping yet.

The Windows token integration method doesn't really help you with the web services integration, as it doesn't change how the log in behaves. It really just affects how the security context for the authenticated user is generated on the server. This should really be an implementation detail for the server based on how it needs to work. I recommend you avoid using token-based integration unless you really really need it though as it limits your flexibility. Claims-based integration is the way to go if your options are open.

If you wanted to try to make something work from a Windows forms client using ADFS V1, you need to find a way to hack the login process to the federation server before you make a web services call. Essentially, you would need the ADFS_WebSsoAuth cookie(s) and append them to your web service proxy before making the calls.

To do that kind of hacking, your forms app would likely need to authenticate with the federation server and execute the ADFS form POST that takes the SAML token issued by the federation server and gets the resulting cookies issued by the server. I think you actually have to do this twice, but it might only be once if there is a single federation server involved. Basically, you just need to reverse engineer the HTTP traffic done by the ADFS login and recreate that using the HttpWebRequest class. It will be icky, but doable.

If you can avoid doing this, that would be even better. You can make this work, but it is a hack and will potentially be hard to support. ADFS V1 just isn't designed to support this use case.

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>

--"CJ" <conorjgallag...@xxxxxxxxxx> wrote in message

news:1176986462.122148.268590@xx

Re: ADFS Development Issues

Hi,

I am looking for some advice on how to develop a certain type of ADFS Authenticating application. Let me try explain how the application must work! I will simplify it for the purpose of this post.

Firstly, we have a client – windows xp (and possibly some vista)
We have a Web Server – Windows server 2003 R2 EE
And a Federation server – Windows server 2003 R2 EE
The Client (Windows XP) will be running a Windows Forms application.
The Windows Forms application is developed using VB.Net and .NET Framework 3.0. (Some of this is C#, so feel free to give C# examples.
All help/advice appreciated)
This windows forms application will communicate to the webserver by means of webservice.
The webservice on the webserver need to be secured using ADFS.

I have successfully got a Claim Aware web application working using the ADFS Step by Step guide. I haven't got the Windows Token side working as the ADFS step by step guide describes this side using sharepoint.

I also got a web service successfully authenticating using adfsas well. Basically, on the web server I pretty much just made a copy of the sample claim app and published the web service files into the same web application. Now when we navigate to our service.asmx file we get asked for authentication details. This works exactly how I expect it to do, but it's using the claims aware method.

The problem is that this won't work with windows forms. The first problem, that I got around, was that I couldn't create a web reference to the web service because the address was changing when I was navigating to the service.asmx. I got around this by temporarily removing the HTTPModule in the web config, creating the reference, and putting the HTTP Module back in. The second problem I had was when I create a new webservice object in windows forms app it doesn't display a logon screen (which I expected to happen). This obviously causes

Re: ADFS Development Issues

trust errors when trying to call the web methods. So basically I need to figure out how to "pre-authenticate" and send through our credentials... or something to that effect.... to allow us to call our web services.

I'm thinking we may have to go down the Windows Token route to do this? But I'm new to this so don't quite know where to start. It seems that claims-aware applications seem to automatically request for auth details which is not quite what we want. We need something with a little more functionality that will allow us to control the authentication process. For example, one of the features we require is to prompt the user to change their password on first sign in if their account is set up to do so. Another feature we require is the ability for the windows application to pass in user logon info without user intervention. The reason for this is that we have a third party web app that will already have authenticated the user and will pass us their login information. So we will not want the user to have to authenticate twice. This means claims aware won't work because we can't preauthenticate to stop the logon screen appearing.

Receiving the third party logon info is all working up to the point where I need to authenticate the details with ADFS. So, ADFS is where I am stuck!

So, from what we can gather from my own research, claims aware applications will not provide us with what we need so I need to consider what functionality Windows Tokens will provide. But most of the examples out there I have found are using sharepoint. Again because I'm new to this I'm finding it hard to use these guides to figure out what I need to do.

So, a summary on what our system needs to do is:

Form app has it's own logon screen (I presume? I don't think there is anyother way to do this?)

Windows forms application gets user details either from logon screen or third party web application.

Forms app sets up Credentials/Auth Token (OR possibly tells the web server to setup the credential/auth token?)

Forms app creates web service object (passes though token/credentials etc or how does this work?)

Re: ADFS Development Issues

Forms app can repeatedly create web services and run web methods without having to re-authenticate

So, I suppose the main question is what is the ideal solution to this scenario using ADFS?

If I need to take a windows token based route then any advice and guidance is much appreciated. Even if you could supply me with some good links or examples that would help me get started on this it would get me moving.

Or, the question I'm afraid to ask, can this be done at all!!!

Thanking you in advance!

Conor. – Hide quoted text –

– Show quoted text –