

microsoft.public.dotnet.security: Re: User.IsInRole is always FALSE

Re: User.IsInRole is always FALSE

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2005-06/0123.html>

From: Joseph Bittman MCAD (*RyanBittman_at_msn.com*)

Date: 06/09/05

Date: Thu, 9 Jun 2005 08:46:08 -0700

June 9, 2005

:-) That helps a lot. Thread.CurrentPrincipal will return whatever account the thread is running under. Since this is a web application, it would be the thread that the web Application is running under; not the client. By default this would be ASPNET on IIS 5.0. This would still be the case unless there are impersonation settings or if you specified your web application to run under a different account in IIS. It sounds like there is impersonation of an Administrator account and therefore doesn't sound like you are checking what the client's roles are. Use the User object to check whether the USER is in a role. The difference is that the currentprincipal object is what the application running under and the User is what the Client web browser is running under. This assumes that you only have Windows Authentication in IIS. This should work now.... As far as the username format, you should step through the code and see what the username looks like. It can be LocalMachineName\Username, or Domain\Username, or BUILTIN\Username. Therefore to always get it right, use the Environment.UserDomainName string. This will return either the machine name or the domain name, depending on what it expects. This way, you don't have to guess which name to put. So try

```
User.IsInRole(@Environment.UserDomainName & "\\TestGroup");
```

I'm not a C# programmer, so you might have to adjust my statement slightly!

:-) You may want to step through the code if it doesn't work and find out what name is actually being returned in User. :-) Let me know how this works!!!

--

Joseph Bittman

Microsoft Certified Application Developer

Web Site: <http://71.35.110.42>

Dynamic IP -- Check here for future changes

"Dominick Baier [DevelopMentor]" <dbaier@pleasepleasenosspamdevelop.com>

wrote in message news:469621632539081378571528@news.microsoft.com...

> Hello Oliver,

>

> ok -

> a) have you disabled anonymous access for the asp.net app?

> b) WindowsIdentity.GetCurrent() gives you the worker process identity - or

> the client identity (if impersonating)

Re: User.IsInRole is always FALSE

microsoft.public.dotnet.security: Re: User.IsInRole is always FALSE

```
> c) better - always use Page/Context.User if you want to make checks
> against the client identity.
>
>
> string user = Page.User.Identity.Name;
>
> bool isInTestgroup2 = Page.User.IsInRole(@"DOMAIN\Testgroup2");
>
> does that make a difference??
>
> -----
> Dominick Baier - DevelopMentor
> http://www.leastprivilege.com
>
>> Hi Joseph,
>>
>> I executed the following code for testing the stuff --> all booleans
>> were false except the local admin groups...
>>
>> WindowsPrincipal windowsPrincipal =
>> (WindowsPrincipal)Thread.CurrentPrincipal;
>> IPrincipal iPrincipal = Thread.CurrentPrincipal;
>>
>> WindowsIdentity windowsIdentity =
>> System.Security.Principal.WindowsIdentity.GetCurrent();
>> string user = windowsIdentity.Name;
>>
>> bool isLocalAdmin =
>> windowsPrincipal.IsInRole(WindowsBuiltInRole.Administrator);
>> bool isInTestgroup1 = windowsPrincipal.IsInRole(@"DOMAIN\Testgroup1");
>>
>> bool isInTestgroup2 = windowsPrincipal.IsInRole(@"DOMAIN\Testgroup2");
>>
>> bool isLocalAdmin_2 = iPrincipal.IsInRole(@"BUILTIN\Administrators");
>>
>> bool isInTestgroup1_2 = iPrincipal.IsInRole(@"DOMAIN\Testgroup1");
>>
>> bool isInTestgroup2_2 = iPrincipal.IsInRole(@"DOMAIN\Testgroup2");
>>
>> Thanks for your help...
>>
>> Oliver Rainer
>>
>> "Joseph Bittman MCAD" <RyanBittman@msn.com> wrote in message
>> news:OJWiWGEbFHA.3032@TK2MSFTNGP10.phx.gbl...
>>
>>> June 9, 2005
>>>
>>> :-) Yes, the BUILTIN will only work for local groups. AD groups are
>>> somewhat more mysterious and have had problems with them myself. When
>>> you call IsInRole, where is the IPrincipal coming from? Are you using
>>> the interface directly, or using an object taken from the
>>> System.Threading.Thread.CurrentPrincipal, or the User object???? This
>>>
>> makes
>>
>>> a big difference in the result that you will get, because I believe
>>> in
>>>
>> your
>>
>>> case that only a principal taken from User will reflect the user's
```

microsoft.public.dotnet.security: Re: User.IsInRole is always FALSE

```
>>>
>> identity
>>
>>> and not ASPNET. If you could post how you got Iprincipal.IsInRole,
>>> then I believe that I can help you further! :-) Have a great day and
>>> until
>>>
>>> later!!!
>>
>>> --
>>> Joseph Bittman Beta ID# 678424
>>> Microsoft Certified Application Developer
>>> Web Site: http://71.35.110.42
>>> Dynamic IP -- Check here for future changes
>>> "Oliver Rainer" <olrainer@tiscali.ch> wrote in message
>>> news:O$%23NC1$aFHA.1044@TK2MSFTNGP10.phx.gbl...
>>>
>>>> Thanks for your suggestion... that really works for local groups
>>>> (BUILTIN),
>>>> but IIS still has the problem to resolve AD groups
>>>> (OURDOMAIN\...)...
>> The
>>
>>>> strange thing is that on certain machines (our developer machines),
>>>> IIS
>>>> can
>>>> succesfully determine if a user belongs to an AD group...
>>>> Do you have other suggestions??
>>>>
>>>> Thanks
>>>> Oliver Rainer
>>>> "Joseph Bittman MCAD" <RyanBittman@msn.com> wrote in message
>>>> news:eIRT4F4aFHA.3280@TK2MSFTNGP12.phx.gbl...
>>>>
>>>>> June 7, 2005
>>>>>
>>>>> hehe I actually have to go to a meeting right now, but a quick
>>>>> suggestion
>>>>> before I leave. Try switching the PrincipalPermission
>>>>> role="domain\groupname" to "BUILTIN\GROUPNAME". There is a little
>>>>> more
>> to
>>
>>>>> this change, but it should work.... depending on which role it is,
>>>>> so
>>>>>
>> if
>>
>>>>> you
>>>>>
>>>>> could post it, that would really help... until later! :-)
>>>>>
>>>>> --
>>>>> Joseph Bittman
>>>>> Microsoft Certified Application Developer
>>>>> Web Site: http://71.35.110.42
>>>>> Dynamic IP -- Check here for future changes
>>>>> "Oliver Rainer" <olrainer@tiscali.ch> wrote in message
>>>>> news:ubg7WH3aFHA.2440@TK2MSFTNGP10.phx.gbl...
>>>>>
>>>>>> Hi, I have the following problem...
>>>>>>
```

microsoft.public.dotnet.security: Re: User.IsInRole is always FALSE

```
>>>>> Pre-requisites:
>>>>> Installation of an Asp.net webservice on a IIS5 server (win2k).
>>>> Anonymous
>>>>
>>>>> access is not allowed, only using windows authentication (intranet
>>>>> app),
>>>>> webmethoda are using the Principalpermssion attribute
>>>>> ([PrincipalPermission(SecurityAction.Demand,
>>>>> Role=@"DomainName\WindowsGroup...")) for identification.
>>>>> Result --> does not work
>>>>>
>>>>> When I change the to [PrincipalPermission(SecurityAction.Demand,
>>>>> name=@"DomainName\Windowsuser...")] -> it runs fine
>>>>>
>>>>> --> conclusion: IIS cannot determine if a user belongs to an AD
>>>>>
>>>> group -->
>>>>
>>>>> IPrincipal.IsInRole(@"DomainName\WindowsGroup") is always FALSE!
>>>>>
>>>>> I already found an article on the web
>>>>> (http://www.kbalertz.com/kb\_894432.aspx --> web dir is UNC
>>>>> share),
>>>>> but
>>>>> didn't solved the problem...
>>>>> Does anyone know more about this issue?
>>>>>
>>>>> Thanks
>>>>> Oliver Rainer
>
>
>
```