

Re: Windows authentication for web service client??

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet.webservices/2005-04/msg001>

- *From:* Keith Elder <keith@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 18 Apr 2005 22:24:52 -0400
-

If you are posting to WEBDAV it is my understanding that you cannot use Integrated authentication since you HAVE to pass it the username and the password in the network credentials. I was going to try to write some information to user's calendars and saw several articles on doing it via WEBDAV. However, you have to ask them for their password and pass it along. This makes it totally useless as far as I am concerned.

If someone knows a way to not have to pass the password through that would be great but I haven't seen anything on how to do it anywhere.

-Keith

solex wrote:

Kevin,

I appreciate your response.

I guess what I am saying here is that it is not working as advertised. I must put together a sample example, but for some reason the users credentials are lost when making the WebDAV request. I get a 401 unauthorized error.

Thanks,
Dan

Re: Windows authentication for web service client??

"Kevin Yu" <koo9@xxxxxxxxxxxxxx> wrote in message
<news:eOY0VXERFHA.4028@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Dan

The bottom line is when enable integrated windows authentication for a service (web app, web service etc) the client need to supply proper credential to the service. now as I mention, DefaultCredentials will always return the credential that the client is running under. so by default, the web service is running ASPNET account. you can however config the web service(I assume that's the client) to run under a different account.

I am not sure what you mean "users id" here, if you mean the login users, then you can set the impersonate=true in the web.config file. so that calls to the WebDAV will use the login users' credentials.

HTH

Kevin

"solex" <solex@xxxxxxxxxxxxxx> wrote in message
<news:%2373WxyDRFHA.2736@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Kevin,

My problem is that the DefaultCredentials is NOT working. If I hard code the credentials using my uid/password and domain it works fine as shown in my first example.

Re: Windows authentication for web service client??

Ideally I want the web service and a subsequent call to Exchange (via WebDAV) to run completely under the users id.

Thanks,
Dan

"Kevin Yu" <koo9@xxxxxxxxxxxx> wrote in message
news:etWV4kCRFHA.508@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

"solex" <solex@xxxxxxxxxxxx> wrote in message
news:%23wMk7BdOFHA.3076@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Kevin,
Thanks for responding, if you (or anyone) sees anything obvious

wrong

with the below summary please let me know.

Thanks,
Dan

I have the following settings

```
Web config:  
  <authentication mode="Windows" />  
  <identity impersonate="true" />
```

IIS:
 Anonymous access has been disabled and Integrated Security

the

only access that is enabled.

Re: Windows authentication for web service client??

Client:

When calling the web service I make sure that I am passing defaultCredentials from the CredentialCache.

I hardcoded a credential using the following code and it works

```
Dim Response As System.Net.HttpWebResponse
Dim Request As HttpWebRequest = CType(WebRequest.Create(URI),
HttpWebRequest)
Dim MyCredentialCache = New System.Net.CredentialCache
MyCredentialCache.Add(New System.Uri(URI), "NTLM", _
    New System.Net.NetworkCredential("myUserID", "myPassword",
"myDomain"))
```

```
Request.Credentials = MyCredentialCache
```

```
make my http WEBDAV request here ...
```

```
Return (Response)
```

But this does not work:

```
Dim Response As System.Net.HttpWebResponse
Dim Request As HttpWebRequest = CType(WebRequest.Create(URI),
HttpWebRequest)
```

```
Request.Credentials = CredentialCache.DefaultCredentials
make my http WEBDAV request here ...
```

```
Return (Response)
```

Re: Windows authentication for web service client??

ok. CredentialCache.DefaultCredentials will return the credentials that the client is running under.
so it doesn't matter what you set before the line:

```
Request.Credentials = CredentialCache.DefaultCredentials
```

it will always return the default credential for the request, but in the working code, since you set credentials in the credentialscache for that particular request URI, so that when the client making calls to the destined service, it will use that credential for the request, that's why it works.

Nor does this:

```
Dim impersonationContext As  
System.Security.Principal.WindowsImpersonationContext  
Dim currentWindowsIdentity As
```

```
System.Security.Principal.WindowsIdentity
```

```
currentWindowsIdentity = CType(mobjUser.Identity,  
System.Security.Principal.WindowsIdentity)  
impersonationContext = currentWindowsIdentity.Impersonate()
```

```
Request.Credentials = CredentialCache.DefaultCredentials  
Dim Response As System.Net.HttpWebResponse  
Dim Request As HttpWebRequest = CType(WebRequest.Create(URI),  
HttpWebRequest)
```

Re: Windows authentication for web service client??

```
Request.Credentials = CredentialCache.DefaultCredentials
```

```
make my http WEBDAV request here ...
```

```
impersonationContext.Undo()
```

```
Return (Response)
```

I have try the same approach using implicit impersonation, what you are doing here is the same as using this line: `Request.Credentials = CredentialCache.DefaultCredentials` since you use this call to get the current identity: `currentWindowsIdentity = CType(mobjUser.Identity,`

```
System.Security.Principal.WindowsIdentity),  
then you do this:
```

```
Request.Credentials = CredentialCache.DefaultCredentials  
thus in fact you are doing the same thing twice.
```

it seems that doing impersonation won't change the default credential, `Request.Credentials = CredentialCache.DefaultCredentials` will always return the credentials that the client is running under as I mentioned above.

I use this code from msdn to do impersonation:

Re: Windows authentication for web service client??

```
#region Public Methods

public bool ImpersonateValidUser()

{

    WindowsIdentity tempWindowsIdentity;

    IntPtr token = IntPtr.Zero;

    IntPtr tokenDuplicate = IntPtr.Zero;

    if(RevertToSelf())

    {

        if(LogonUserA(_userName, _domain, _password, LOGON32_LOGON_INTERACTIVE

LOGON32_PROVIDER_DEFAULT, ref token) != 0)

        {

            if(DuplicateToken(token, 2, ref tokenDuplicate) != 0)

            {
```

Re: Windows authentication for web service client??

```
tempWindowsIdentity = new WindowsIdentity(tokenDuplicate);

impersonationContext = tempWindowsIdentity.Impersonate();

if (impersonationContext != null)

{

    CloseHandle(token);

    CloseHandle(tokenDuplicate);

    return true;

}

}

}

}

}

if(token!= IntPtr.Zero)

    CloseHandle(token);

if(tokenDuplicate!=IntPtr.Zero)
```

Re: Windows authentication for web service client??

```
CloseHandle(tokenDuplicate);

return false;

}

//reverse the security context

public void UndoImpersonation()

{

if(impersonationContext!=null)

impersonationContext.Undo();

}

#endregion

#region Win32 calls

[DllImport("advapi32.dll")]

private static extern int LogonUserA(String lpszUserName,
```

Re: Windows authentication for web service client??

```
String lpszDomain,  
  
String lpszPassword,  
  
int dwLogonType,  
  
int dwLogonProvider,  
  
ref IntPtr phToken);  
  
[DllImport("advapi32.dll", CharSet=CharSet.Auto, SetLastError=true)]  
  
private static extern int DuplicateToken(IntPtr hToken,  
  
int impersonationLevel,  
  
ref IntPtr hNewToken);  
  
[DllImport("advapi32.dll", CharSet=CharSet.Auto, SetLastError=true)]  
  
private static extern bool RevertToSelf();  
  
[DllImport("kernel32.dll", CharSet=CharSet.Auto)]  
  
private static extern bool CloseHandle(IntPtr handle);
```

Re: Windows authentication for web service client??

```
#endregion
```

```
}
```

in conclusion, only when the correct credential in the credentialsCache
for
that
request (that particular URI), it request have access permission.

thanks for your code. I will give it a try.

"Kevin Yu" <koo9@xxxxxxxxxxxx> wrote in message
news:u0yUSScOFHA.580@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

"solex" <solex@xxxxxxxxxxxx> wrote in message
news:%23sSDjOSOFHA.244@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

I'm having a similar problem

I have a web service that make a webDav request to

I
have
impersonation
on
but
when
I

Re: Windows authentication for web service client??

use
the
defaultCredentials
in
the

web

services
to
make
the
webdav
request
I
get
an
Unauthorized
401
error.

My

credentials have rights to make this request and I
trying
to figure it out.

The
service
works
if
I
hard
code
my
Network
credentials
in
the

service

but

Re: Windows authentication for web service client??

does not otherwise.

Hardcoded into your code? create a credential instead of defaultcredentials?

I thought one can only create credential for "basic" or "digest" authentication mode.

I try implicit impersonation, it won't work, even if you are impersonating, the web service has to put the credential on the soap message in order for it to be authenticated, because that's what all the hosting services see when interacting with each other. I want to do explicit impersonation.

in .net 2.0, there will be a better support or even WSE 2.0, but this is not my options here. since if we were to use WSE 2.0, there will be a long process of

paper

Re: Windows authentication for web service client??

work
and testing and questioning.....

Any help with this would also be appreciated.

Thanks,
Dan

"Kevin Yu" <koo9@xxxxxxxxxxxx> wrote in message
news:eOariLKOFHA.1476@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

but
the
problem
with
impersonation
in
the
code
is
after

LogonUser()

win32

call, will the defaultcredentials be set to
then?

Re: Windows authentication for web service client??

"Kevin Yu" <koo9@xxxxxxxxxxxx> wrote in message
news:OEbaAMIOFHA.2356@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

I think impersonation will do , enable
specified
the user, use code call the web service
username/password.

"Brock Allen" <ballen@xxxxxxxxxxxxxxxxxxxx>
news:453919632490103600068528@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

The
ASPNET
account
is
a
local
account,
so
the
other
machine
or

domain

wouldn't
know
about
it.
You
can

Re: Windows authentication for web service client??

either
run
you
web
app
under
a

different

account,
but
that
affects
the
rest
of
the
code
in
there
too.
The
other
approach
is
to
have
a
dedicated
account
(instead
of
using
the

current

identity
of
ASPNET)
that
you
can
use

Re: Windows authentication for web service client??

to
do
the
authentication

and

then
use those credentials from the

-Brock
DevelopMentor
<http://staff.develop.com/balle>

hi all

got
a
question
here,
a
web
service
secure
mode
is
set
to

"windows",

on

the client side

Re: Windows authentication for web service client??

when supplying the crede

somewbservice.Authentic

somewbservice.Authentic

ssoAuth.PreAuthenticate

ssoAuth.Credentials

=

```
System.Net.CredentialCache.DefaultCredentials;
```

from the info here

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref>

/html/frlrfSystemNetCredentialCacheClassDefaultCredentialsTopic.asp

the
defaultcredential
should
supply
the
current
security

context

that
the
client

Re: Windows authentication for web service client??

is
running,
but
in
my
case
the
client
is
another
web
service
running

on
another
server,
now
by
default
the
account
that
the

client(the

calling web service) is

so
on
the
host(somewebsevice
I
should
add
the

clientdomain\ASPNET

account into the windows

Re: Windows authentication for web service client??