

friendly error messages for usernameForCertificateSecurity

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.webservices.enhancements/2006-07/>

- *From:* geobier <geobier@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 11 Jul 2006 11:36:01 -0700
-

This has been asked several times on the web, but I have yet to see a solution.

In short:

Using usernameForCertificateSecurity, with a custom UsernameTokenManager, AuthenticateToken detects an error ("Invalid user id or password", "account is locked", "account expires in 5 days"). How can this error message be returned to the client in a simple way.

More specifically:

```
protected override string AuthenticateToken(UsernameToken token)
{
    .....

    // password is invalid!!!!
    // What is the code to put here????
    // throw new SoapException("Bad juju", new XmlQualifiedName("geb:hi"));
}
```

In the code above if I comment out the throwing of the soap exception on the client side, I receive back the dreaded:

"WSE910: An error happened during the processing of a response message, and you can find the error in the inner exception."

the inner exception is equally useless:

"Security requirements are not satisfied because the security header is not present in the incoming message."

If you recast the outer exception as follows:

```
catch (Exception exc)
{
    Microsoft.Web.Services3.ResponseProcessingException temp =
    (Microsoft.Web.Services3.ResponseProcessingException) exc;
```

friendly error messages for usernameForCertificateSecurity

```
string message = temp.Response.Fault.Message;  
}
```

you CAN get to the original message, ("bad juju") just follow the ----> three deep:

message:

```
System.Web.Services.Protocols.SoapHeaderException: Server unavailable,  
please try later ----> System.ApplicationException: WSE841: An error occurred  
processing an outgoing fault response. ---->  
System.Web.Services.Protocols.SoapException:  
System.Web.Services.Protocols.SoapException: Bad juju\r\n at  
Ingenix.Omx.WS.UTM.AuthenticateToken(UsernameToken token) in  
C:\\WebSites\\HelloWSE\\TokenManager\\TokenManager.cs:line 200\r\n at  
Microsoft.Web.Services3.Security.Tokens.UsernameTokenManager.VerifyToken(SecurityToken  
token)\r\n at Ingenix.Omx.WS.UTM.VerifyToken(SecurityToken token) in  
C:\\WebSites\\HelloWSE\\TokenManager\\TokenManager.cs:line 169\r\n at  
Microsoft.Web.Services3.Security.Tokens.SecurityTokenManager.LoadXmlSecurityToken(XmlElement  
element)\r\n at  
Microsoft.Web.Services3.Security.Tokens.SecurityTokenManager.GetTokenFromXml(XmlElement  
element)\r\n at  
Microsoft.Web.Services3.Security.Security.LoadToken(XmlElement element,  
SecurityConfiguration configuration, Int32& tokenCount)\r\n at  
Microsoft.Web.Services3.Security.Security.LoadXml(XmlElement element)\r\n at  
Microsoft.Web.Services3.Security.Security.CreateFrom(SoapEnvelope  
envelope, String localActor, String serviceActor)\r\n at  
Microsoft.Web.Services3.Security.ReceiveSecurityFilter.ProcessMessage(SoapEnvelope  
envelope)\r\n at  
Microsoft.Web.Services3.Pipeline.ProcessInputMessage(SoapEnvelope  
envelope)\r\n at  
Microsoft.Web.Services3.WseProtocol.FilterRequest(SoapEnvelope  
requestEnvelope)\r\n at  
Microsoft.Web.Services3.WseProtocol.RouteRequest(SoapServerMessage  
message)\r\n at  
System.Web.Services.Protocols.SoapServerProtocol.Initialize()\r\n at  
System.Web.Services.Protocols.ServerProtocolFactory.Create(Type type,  
HttpContext context, HttpRequest request, HttpResponse response, Boolean&  
abortProcessing)\r\n ---- End of inner exception stack trace ----\r\n ----  
End of inner exception stack trace ----" string
```

Trying to tell my clients that they need to parse this message to get the true error message is a less than desirable solution.

Questions:

- 1) Is there some way to have the exception show up in the InnerMessage exception tree? Telling my clients to follow the inner exception tree is a much better solution.
- 2) Is there an alternative way of communicating the error information back

friendly error messages for usernameForCertificateSecurity

rather than throwing an exception. The above example uses a SoapException, if you change it to throw new Exception("bad juju"), the results are similar. Anyone got a better solution?

Restrictions:

Custom assertion policy on the client side is not going to be an acceptable solution with my clients. We are already pushing the limits on client configuration complexity using the canned policies.

On the server side, which I control, I am willing to code as complicated a solution as required to be able to return to my clients a reasonable message.

Observation:

This seems like a common use case for this security case. All the quickstarts and examples I was able to track down skip this issue. I only found one client side example that mentioned this case and it simply showed the client catching the exception, without actually going into what the exception contained. Is anyone actually really using this stuff (sigh)? As I said, other people have asked this question before.

Alternative:

One thought I have been playing with is always accepting the credential, but setting a flag in the UsernameToken indicating that the user did not validate, with a reason code. My server side protected webmethods would then do a validation check at entry, and at that point I could return a reasonable error message or SoapException. This is one heck of an ugly workaround, but gets me out of this mess. Opinions?

thanks,
--george

--
thanks,
--george
.