

WSE402: The message does not conform to the policy it was mapped t

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.webservices.enhancements/2006-08/>

- *From:* Chris Fink <ChrisFink@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 28 Aug 2006 08:53:01 -0700
-

I am receiving the following error message from my client when consuming a WSE 2 SP3 webservice that is requiring client side certs and username tokens:

WSE402: The message does not conform to the policy it was mapped to. at Microsoft.Web.Services2.Policy.SimplePolicyVerifier.VerifyMessageWithExpression(PolicyExpression expression, SoapEnvelope message, EndpointReference endpoint, String action, Uri requestEndpoint) at Microsoft.Web.Services2.Policy.SimplePolicyVerifier.Verify(SoapEnvelope message) at Microsoft.Web.Services2.Policy.PolicyVerificationInputFilter.ProcessMessage(SoapEnvelope envelope) at Microsoft.Web.Services2.Pipeline.ProcessInputMessage(SoapEnvelope envelope) at Microsoft.Web.Services2.InputStream.GetRawContent() at Microsoft.Web.Services2.InputStream.get_Length() at System.Xml.XmlScanner..ctor(TextReader reader, XmlNameTable ntable) at System.Xml.XmlTextReader..ctor(String url, TextReader input, XmlNameTable nt) at System.Xml.XmlTextReader..ctor(TextReader input) at System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse(SoapClientMessage message, WebResponse response, Stream responseStream, Boolean asyncCall) at System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodName, Object[] parameters) at ApplicationMessagingWS.Dispatch(String messageType, String correlationId, String messageBody, String userName, String applicationName, String instance, String postBackUrl) at DellWSE2SP3.ConsumeDellMSS.Dispatch(String messageType, String correlationId, String messageBody, String userName, String password, String applicationName, String instance, String postBackUrl) Microsoft.Web.Services2

I believe the error to be that I am missing the policy section for username tokens; I used the WSE wizard to create the policy and selected the client side certs option not knowing how to select both a client side cert and username token.

Following is my web.config:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<configSections>
<section name="microsoft.web.services2"
```

WSE402: The message does not conform to the policy it was mapped t

```
type="Microsoft.Web.Services2.Configuration.WebServicesConfiguration,  
Microsoft.Web.Services2, Version=2.0.0.0, Culture=neutral,  
PublicKeyToken=31bf3856ad364e35" />
```

```
</configSections>
```

```
<system.web>
```

```
<!-- DYNAMIC DEBUG COMPILATION
```

Set compilation debug="true" to enable ASPX debugging. Otherwise, setting this value to

false will improve runtime performance of this application.

Set compilation debug="true" to insert debugging symbols (.pdb information)

into the compiled page. Because this creates a larger file that executes

more slowly, you should set this value to true only when debugging and to

false at all other times. For more information, refer to the documentation about

debugging ASP.NET files.

```
-->
```

```
<compilation defaultLanguage="c#" debug="true" />
```

```
<!-- CUSTOM ERROR MESSAGES
```

Set customErrors mode="On" or "RemoteOnly" to enable custom error messages, "Off" to disable.

Add <error> tags for each of the errors you want to handle.

"On" Always display custom (friendly) messages.

"Off" Always display detailed ASP.NET error information.

"RemoteOnly" Display custom (friendly) messages only to users not running

on the local Web server. This setting is recommended for security purposes, so

that you do not display application detail information to remote clients.

```
-->
```

```
<customErrors mode="RemoteOnly" />
```

```
<!-- AUTHENTICATION
```

This section sets the authentication policies of the application.

Possible modes are "Windows",

"Forms", "Passport" and "None"

"None" No authentication is performed.

"Windows" IIS performs authentication (Basic, Digest, or Integrated Windows) according to

its settings for the application. Anonymous access must be disabled in IIS.

"Forms" You provide a custom form (Web page) for users to enter their credentials, and then

you authenticate them in your application. A user credential token is stored in a cookie.

"Passport" Authentication is performed via a centralized authentication service provided

WSE402: The message does not conform to the policy it was mapped t

by Microsoft that offers a single logon and core profile services for member sites.

-->

```
<authentication mode="Windows" />
```

```
<!-- AUTHORIZATION
```

This section sets the authorization policies of the application.

You can allow or deny access

to application resources by user or role. Wildcards: "*" mean

everyone, "?" means anonymous

(unauthenticated) users.

-->

```
<authorization>
```

```
<allow users="*" />
```

```
<!-- Allow all users -->
```

```
<!-- <allow users="[comma separated list of users]"
```

```
roles="[comma separated list of roles]"/>
```

```
<deny users="[comma separated list of users]"
```

```
roles="[comma separated list of roles]"/>
```

-->

```
</authorization>
```

```
<!-- APPLICATION-LEVEL TRACE LOGGING
```

Application-level tracing enables trace log output for every page within an application.

Set trace enabled="true" to enable application trace logging. If

pageOutput="true", the

trace information will be displayed at the bottom of each page.

Otherwise, you can view the

application trace log by browsing the "trace.axd" page from your

web application

root.

-->

```
<trace enabled="false" requestLimit="10" pageOutput="false"
```

```
traceMode="SortByTime" localOnly="true" />
```

```
<!-- SESSION STATE SETTINGS
```

By default ASP.NET uses cookies to identify which requests belong to a particular session.

If cookies are not available, a session can be tracked by adding a

session identifier to the URL.

To disable cookies, set sessionState cookieless="true".

-->

```
<sessionState mode="InProc"
```

```
stateConnectionString="tcpip=127.0.0.1:42424" sqlConnectionString="data
```

```
source=127.0.0.1;Trusted_Connection=yes" cookieless="false" timeout="20" />
```

```
<!-- GLOBALIZATION
```

This section sets the globalization settings of the application.

-->

```
<globalization requestEncoding="utf-8" responseEncoding="utf-8" />
```

```
</system.web>
```

```
<microsoft.web.services2>
```

```
<security>
```

```
<x509 storeLocation="LocalMachine" allowTestRoot="true"
```

WSE402: The message does not conform to the policy it was mapped t

```
allowRevocationUrlRetrieval="false" />
</security>
<diagnostics>
<trace enabled="true" input="InputTrace.webinfo"
output="OutputTrace.webinfo" />
<policyTrace enabled="true" input="ReceivePolicy.webinfo"
output="SendPolicy.webinfo" />
</diagnostics>
</policy>
<cache name="policyCache.config" />
</policy>
</microsoft.web.services2>
</configuration>
```

Following is my policy:

```
<?xml version="1.0" encoding="utf-8"?>
<policyDocument xmlns="http://schemas.microsoft.com/wse/2003/06/Policy>
<mappings xmlns:wse="http://schemas.microsoft.com/wse/2003/06/Policy>
<!--The following policy describes the policy requirements for the
service: http://customerxxx/mss/webservices/ApplicationMessagingWS.asmx.-->
<endpoint
uri="http://customerxxx/mss/webservices/ApplicationMessagingWS.asmx>
<defaultOperation>
<request policy="#Sign-X.509-Encrypt-X.509-2" />
<response policy="#Sign-X.509-Encrypt-X.509-3" />
<fault policy="" />
</defaultOperation>
</endpoint>
</mappings>
<policies
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
xmlns:wsp="http://schemas.xmlsoap.org/ws/2002/12/policy
xmlns:wssp="http://schemas.xmlsoap.org/ws/2002/12/secext
xmlns:wse="http://schemas.microsoft.com/wse/2003/06/Policy
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing>
<wsp:Policy wsu:Id="Sign-X.509-Encrypt-X.509-2">
<!--MessagePredicate is used to require headers. This assertion should
be used along with the Integrity assertion when the presence of the signed
element is required. NOTE: this assertion does not do anything for
enforcement (send-side) policy.-->
<wsp:MessagePredicate wsp:Usage="wsp:Required"
Dialect="http://schemas.xmlsoap.org/2002/12/wsse#part>wsp:Body()
wsp:Header(wsa:To) wsp:Header(wsa:Action) wsp:Header(wsa:MessageID)
wse:Timestamp()</wsp:MessagePredicate>
<!--The Integrity assertion is used to ensure that the message is
signed with X.509. Many Web services will also use the token for
authorization, such as by using the <wse:Role> claim or specific X.509
claims.-->
<wssp:Integrity wsp:Usage="wsp:Required">
<wssp:TokenInfo>
```

WSE402: The message does not conform to the policy it was mapped t

WSE402: The message does not conform to the policy it was mapped t

<!--The SecurityToken element within the TokenInfo element describes which token type must be used for Signing.-->

<wssp:SecurityToken>

<wssp:TokenType>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3</w

<wssp:TokenIssuer>CN=Root Agency</wssp:TokenIssuer>

<wssp:Claims>

<!--By specifying the SubjectName claim, the policy system can look for a certificate with this subject name in the certificate store indicated in the application's configuration, such as LocalMachine or CurrentUser. The WSE X.509 Certificate Tool is useful for finding the correct values for this field.-->

<wssp:SubjectName

MatchType="wssp:Exact">CN=ABCHBTServer</wssp:SubjectName>

<wssp:X509Extension OID="2.5.29.14"

MatchType="wssp:Exact">KEYREMOVEDFORSECURITYPURPOSES=</wssp:X509Extension>

</wssp:Claims>

</wssp:SecurityToken>

</wssp:TokenInfo>

<wssp:MessageParts

Dialect="http://schemas.xmlsoap.org/2002/12/wsse#part>wsp:Body()

wsp:Header(wsa:Action) wsp:Header(wsa:FaultTo) wsp:Header(wsa:From)

wsp:Header(wsa:MessageID) wsp:Header(wsa:RelatesTo) wsp:Header(wsa:ReplyTo)

wsp:Header(wsa:To) wse:Timestamp()</wssp:MessageParts>

</wssp:Integrity>

<!--The Confidentiality assertion is used to ensure that the SOAP Body is encrypted.-->

<wssp:Confidentiality wsp:Usage="wsp:Required">

<wssp:KeyInfo>

<!--The SecurityToken element within the KeyInfo element describes which token type must be used for Encryption.-->

<wssp:SecurityToken>

<wssp:TokenType>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3</w

<wssp:TokenIssuer>O=ABC Inc., CN=ABC Inc. Enterprise Utility

CA1</wssp:TokenIssuer>

<wssp:Claims>

<!--By specifying the SubjectName claim, the policy system can look for a certificate with this subject name in the certificate store indicated in the application's configuration, such as LocalMachine or CurrentUser. The WSE X.509 Certificate Tool is useful for finding the correct values for this field.-->

<wssp:SubjectName MatchType="wssp:Exact">C=US, S=TX, L=Austin,

O=ABC Inc., OU=Information Technology, CN=MSS Spore,

E=webfarm@xxxxxxx</wssp:SubjectName>

<wssp:X509Extension OID="2.5.29.14"

MatchType="wssp:Exact">KEYREMOVEDFORSECURITYPURPOSES=</wssp:X509Extension>

</wssp:Claims>

</wssp:SecurityToken>

</wssp:KeyInfo>

<wssp:MessageParts

WSE402: The message does not conform to the policy it was mapped t

WSE402: The message does not conform to the policy it was mapped t

Dialect="http://schemas.xmlsoap.org/2002/12/wsse#part>wsp:Body()</wssp:MessageParts>
</wssp:Confidentiality>
</wsp:Policy>

<wsp:Policy wsu:Id="Sign-X.509-Encrypt-X.509-3">
<!--MessagePredicate is used to require headers. This assertion should
be used along with the Integrity assertion when the presence of the signed
element is required. NOTE: this assertion does not do anything for
enforcement (send-side) policy.-->

<wsp:MessagePredicate wsp:Usage="wsp:Required"
Dialect="http://schemas.xmlsoap.org/2002/12/wsse#part>wsp:Body()
wsp:Header(wsa:To) wsp:Header(wsa:Action) wsp:Header(wsa:MessageID)
wse:Timestamp()</wsp:MessagePredicate>

<!--The Integrity assertion is used to ensure that the message is
signed with X.509. Many Web services will also use the token for
authorization, such as by using the <wse:Role> claim or specific X.509
claims.-->

<wssp:Integrity wsp:Usage="wsp:Required">

<wssp:TokenInfo>

<!--The SecurityToken element within the TokenInfo element
describes which token type must be used for Signing.-->

<wssp:SecurityToken>

<wssp:TokenType>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3</w

<wssp:TokenIssuer>O=ABC Inc., CN=ABC Inc. Enterprise Utility
CA1</wssp:TokenIssuer>

<wssp:Claims>

<!--By specifying the SubjectName claim, the policy system can
look for a certificate with this subject name in the certificate store
indicated in the application's configuration, such as LocalMachine or
CurrentUser. The WSE X.509 Certificate Tool is useful for finding the correct
values for this field.-->

<wssp:SubjectName MatchType="wssp:Exact">C=US, S=TX, L=Austin,
O=ABC Inc., OU=Information Technology, CN=MSS Spore,
E=webfarm@xxxxxxx</wssp:SubjectName>

<wssp:X509Extension OID="2.5.29.14"

MatchType="wssp:Exact">KEYREMOVEDFORSECURITYPURPOSES=</wssp:X509Extension>

</wssp:Claims>

</wssp:SecurityToken>

</wssp:TokenInfo>

<wssp:MessageParts

Dialect="http://schemas.xmlsoap.org/2002/12/wsse#part>wsp:Body()
wsp:Header(wsa:Action) wsp:Header(wsa:FaultTo) wsp:Header(wsa:From)
wsp:Header(wsa:MessageID) wsp:Header(wsa:RelatesTo) wsp:Header(wsa:ReplyTo)
wsp:Header(wsa:To) wse:Timestamp()</wssp:MessageParts>

</wssp:Integrity>

<!--The Confidentiality assertion is used to ensure that the SOAP Body
is encrypted.-->

<wssp:Confidentiality wsp:Usage="wsp:Required">

<wssp:KeyInfo>

<!--The SecurityToken element within the KeyInfo element describes
which token type must be used for Encryption.-->

WSE402: The message does not conform to the policy it was mapped t

WSE402: The message does not conform to the policy it was mapped t

<wssp:SecurityToken>

<wssp:TokenType>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3</w

<wssp:TokenIssuer>CN=Root Agency</wssp:TokenIssuer>

<wssp:Claims>

<!--By specifying the SubjectName claim, the policy system can look for a certificate with this subject name in the certificate store indicated in the application's configuration, such as LocalMachine or CurrentUser. The WSE X.509 Certificate Tool is useful for finding the correct values for this field.-->

<wssp:SubjectName

MatchType="wssp:Exact">CN=ABCHBTServer</wssp:SubjectName>

<wssp:X509Extension OID="2.5.29.14"

MatchType="wssp:Exact">KEYREMOVEDFORSECURITYPURPOSES=</wssp:X509Extension>

</wssp:Claims>

</wssp:SecurityToken>

</wssp:KeyInfo>

<wssp:MessageParts

Dialect="http://schemas.xmlsoap.org/2002/12/wsse#part>wsp:Body()</wssp:MessageParts>

</wssp:Confidentiality>

</wsp:Policy>

</policies>

</policyDocument>

And last, here is a code snippet of my client (which itself is a WebService) that is calling the WSE webservice (note the code to add the usernametoken):

try

{

ApplicationMessagingWS oWS = new ApplicationMessagingWS();

UsernameToken tokenUsername = new UsernameToken(userName, password, PasswordOption.SendPlainText);

oWS.RequestSoapContext.Security.Tokens.Add(tokenUsername);

return oWS.Dispatch(messageType, correlationId, messageBody, userName, applicationName, instance, postBackUrl);

↓

catch (Exception ex)

{

return string.Format("Exception = Y : {0} {1} {2}", ex.Message.ToString(), ex.StackTrace.ToString(), ex.Source.ToString());

↓

Any help is appreciated. I am confident the problem is that I need to modify my policy to allow for client side cert AND username tokens, but am unsure of how to do that.

.