

The message must contain a wsa:To header

## The message must contain a wsa:To header

---

*Source:*

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

---

- *From:* Chris Fink <[ChrisFink@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:ChrisFink@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 13 Jul 2006 13:12:01 -0700
- 

I am using WSE 2 SP3 and receiving the following error from my client calling a secure service using client certificates. When I setup the policy, I required signing and encryption for the request and response.

System.Web.Services.Protocols.SoapException: Server was unable to process request. ----> The message must contain a wsa:To header.

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 applicationName, String instance, String postBackUrl)

This is the policy file

```
<?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://xxx/mss/webservices/ApplicationMessagingWS.asmx .-->
```

```
<endpoint uri="http://xxx/mss/webservices/ApplicationMessagingWS.asmx">
```

```
<defaultOperation>
```

```
<request policy="#Sign-X.509-Encrypt-X.509" />
```

```
<response policy="#Sign-X.509-Encrypt-X.509-1" />
```

```
<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
```

The message must contain a wsa:To header

```
<xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing>  
<wsp:Policy wsu:Id="Sign-X.509-Encrypt-X.509">  
<!--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>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=  
HBTCClient</wssp:SubjectName>  
<wssp:X509Extension OID="2.5.29.14"  
MatchType="wssp:Exact">u11Ev47jqXyrb0gujx/GRPFUrw=</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= Inc., CN= Inc. Enterprise Utility
```

The message must contain a wsa:To header

The message must contain a wsa:To header

```
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= Inc., OU=Information Technology, CN=MSS Spore, E=webfarm@
..com</wssp:SubjectName>
<wssp:X509Extension OID="2.5.29.14"
MatchType="wssp:Exact">rrRD87efOO5bpHFLxT+psuYqMKM=</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>
</wssp:Policy>
<wssp:Policy wsu:Id="Sign-X.509-Encrypt-X.509-1">
<!--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)
wsu: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= Inc., CN= 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= Inc., OU=Information Technology, CN=MSS Spore, E=webfarm@
..com</wssp:SubjectName>
<wssp:X509Extension OID="2.5.29.14"
```

The message must contain a wsa:To header

The message must contain a wsa:To header

```
MatchType="wssp:Exact">rrRD87efOO5bpHFLxT+psuYqMKM=</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>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=  
HBTCClient</wssp:SubjectName>  
<wssp:X509Extension OID="2.5.29.14"  
MatchType="wssp:Exact">u1IEv47jqXyrb0gujx/GRPFUrw=</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 the web.config

```
<?xml version="1.0" encoding="utf-8"?>  
<configuration>  
<configSections>  
<section name="microsoft.web.services2"  
type="Microsoft.Web.Services2.Configuration.WebServicesConfiguration,  
Microsoft.Web.Services2, Version=2.0.0.0, Culture=neutral,  
PublicKeyToken=31bf3856ad364e35" />  
</configSections>  
<system.web>  
<!-- DYNAMIC DEBUG COMPILATION
```

The message must contain a wsa:To header

The message must contain a wsa:To header

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

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.

The message must contain a wsa:To header

The message must contain a wsa:To header

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="CurrentUser" allowTestRoot="true" />

</security>

<diagnostics>

<trace enabled="true" input="InputTrace.webinfo"

output="OutputTrace.webinfo" />

<policyTrace enabled="true" input="ReceivePolicy.webinfo"

output="SendPolicy.webinfo" />

The message must contain a wsa:To header

The message must contain a wsa:To header

```
</diagnostics>  
<policy>  
<cache name="policyCache.config" />  
</policy>  
</microsoft.web.services2>  
</configuration>
```

Any help is appreciated. I have not found much information on the web.

.