XML Signer

This topic explains how to configure and apply the XML Signer tool in SOAtest and Virtualize. This tool signs XML documents for security purposes. Sections include:

- Understanding XML Signature
- Configuring the XML Signer Tool
- Tool Settings
- Usage Notes
- Related Tutorials

Understanding XML Signature

In order to securely send data across the Internet during a Web service transaction, security standards must be put in place to ensure that the different users taking part in the transaction can identify each other. The XML Signature standard recommended by the W3C defines a process that allows any data in XML documents to be digitally signed. With XML Signature, Web service users can verify the identity of others involved in a transaction and can be ensured that the data has not been altered since the document was signed. OASIS leverages this standard so that it can be used in SOAP.

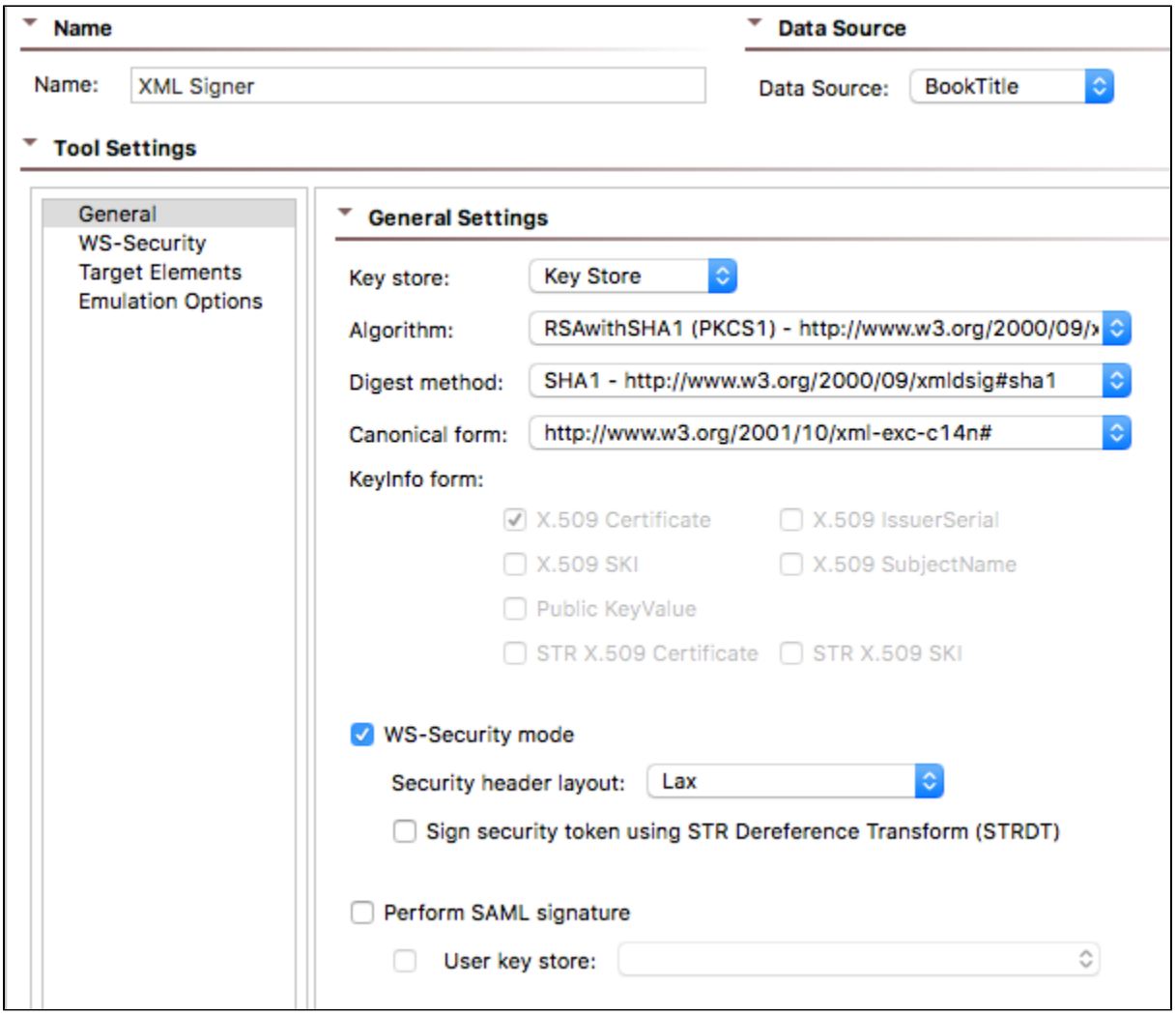
Configuring the XML Signer Tool

The XML Signer Tool supports the W3C XML Signature standard and allows you to digitally sign data to be sent as Web service transactions. The XML Signer Tool also allows you to sign individual elements of the XML document, or the entire document itself. This feature is especially useful for Web service transactions that are performed between multiple partners or endpoints. For example, a transaction for the purchase of car may take place through a Web service. In this instance, the buyer would have to sign certain parts of the document, the loan officer may have to sign certain parts of the document, and the seller would have to sign certain parts of the document.

Tool Settings

The following options display in the left pane of the Tool Settings tab:

- General Settings
- WS-Security Settings
- Target Elements Settings
- Emulation Options Settings
- Input Type Tab



General Settings

The following options are available in the General tab:

Key Store	Select the Key Store used to verify your identity and to sign the XML data from the **Key Store** drop-down menu. The Key Stores available in this menu are dependent on the Key Stores you added at the test or Responder suite level.For more information on adding Key Stores, see Adding Global Key Stores in SOAtest.
Algorithm	Specifies the unique algorithm used for defining the certificate keys.
Digest method	Choose a message digest algorithm for signing the data.
Canonical form	Specifies the algorithm used to create a canonicalized form of the information being signed.

KeyInfo form	KeyInfo is an optional element within the signature; it contains public key information needed to validate the signature. It is only applicable when performing an enveloped signature on POX or SAML (with WS-Security mode off). Select which options should be included within the KeyInfo element. The following options are available: • **X509 Certificate**: Includes a X509Data/X509Certificate element containing the base64-encoded certificate. This is the recommended default. • **X509 IssuerSerial**: Includes a X509Data/X509IssuerSerial element containing the X.509 name/serial number pair. • **X509 SKI**: Includes a X509Data/X509SKI element containing the base64 encoded subject key identifier. The certificate must be X.509 version 3. • **X509SubjectName**: Includes a X509Data/X509SubjectName element containing the X.509 subject name. • **Public KeyValue**: Includes a KeyValue/RSAKeyValue element containing the base64-encoded public key. • **STR X509 Certificate**: Includes a wsse:SecurityTokenReference/wsse:KeyIdentifier element containing the base64-encoded certificate. • **STR X509 SKI**: Includes a wsse:SecurityTokenReference/wsse:KeyIdentifier element containing the base64 encoded subject key identifier.
WS-Security mode	Enables signing SOAP messages according to OASIS WS-Security. The version of the OASIS standard can be configured in the Emulation Options Settings. By default, the content of the entire SOAP body will be automatically signed as configured in the Target Elements Settings. When this mode is disabled, an enveloped signature is performed on target elements. Choose a setting from the **Security header layout** drop-down menu to specify which layout rules to apply when adding items to the WS-Security header. The following settings are available: • **Lax**: Items are added to the security header in any order that conforms to WSS: SOAP Message Security. • **LaxTimestampFirst**: Same as **Lax**, except that the first item in the security header MUST be a wsse:Timestamp. • **LaxTimestampLast**: Same as **Lax**, except that the last item in the security header MUST be a wsse:Timestamp. • **Strict**: Items are added to the security header following the numbered layout rules described below according to a general principle of 'declare before use'. Enable the **Sign security token using STR Dereference Transform (STRDT)** option to automatically include the security token in the signature by referencing a security token reference (STR) using the STR Dereference Transform (STRDT) algorithm. By default, the signature will reference the STR that is automatically generated in the signature under the KeyInfo element. If a different STR should be referenced, such as an STR from the Security header, then include the STR in the Target Elements configuration.
Perform SAML signature	Enable this option to sign a SOAP message based on the OASIS SAML Token profile or to perform an enveloped signature on a SAML assertion. If the WS-Security mode option is disabled, an enveloped SAML signature will be formed on the SAML assertions selected under Target Elements Settings. SAML assertions using the holder-of-key confirmation method should typically have an enveloped signature. If the WS-Security mode option is enabled, the SOAP envelope will be signed based on the OASIS SAML Token profile. Y*ou must choose the correct Emulation Option, otherwise the signature will not be successful:* • For SAML 1.1, select WSS4J 1.5 or later or OASIS 1.1 or later. • For SAML 2, select WSS4J 1.6 or OASIS 1.1.1.
User Key Store	Enable this option to references the requester's keystore. This option is only applicable for SAML assertions using the holder-of-key confirmation method. This option should be disabled when the assertion does not have an enveloped signature, which is typical for sender-vouches confirmation. If the WS-Security mode is disabled, the keystore can optionally be used to add the user's public KeyInfo into the assertion's subject confirmation before performing an enveloped signature on the SAML assertion. This is applicable for the holder-of-key scenario where the assertion holds the user's key and is signed by the assertion issuer. If the WS-Security mode is enabled, the following requirements must be met in order for the signature on the SOAP message to be successful: 1. The holder-of-key assertion must hold the user's certificate referenced by this keystore. 2. The assertion in the SOAP header must already have an enveloped signature. You can chain two XML Signers together to facilitate holder-of-key confirmation. Configure the first signer to add the your public KeyInfo to the assertion's subject confirmation, then perform an enveloped signature on the assertion with the WSS mode off. Next, the second signer can sign the SOAP envelope containing the SAML assertion based on the OASIS SAML token profile with the WSS mode on.

WS-Security Settings

The following options are available in the WS-Security tab:

Form	Choose the appropriate form to specify the key and certificate used for encryption.
Actor	Enter a value to specify a SOAP actor.
Add must Understand=1	Specifies whether or not the receiver must recognize and verify the signature of the message. If this option is enabled, a SOAP fault will be sent back if the receiver does not know how to process the signature header.

Add Timestamp	Enable this option to add a timestamp to the message. The following settings are available: • Enable **Sign timestamp** to provide a digital signature with the timestamp. • Enable **Add expiration** to enter an expiration value.

Target Elements Settings

Enable the **SOAP body/entire document** option to sign the entire SOAP body or entire XML document.

To specify an XPath and sign a specific element within the XML document:

1. Disable the **SOAP body/entire document** option and click the **Tree** tab.
2. Choose an element and click **Extract**. A row will appear in the Selected Element list.
3. Click **Modify** to make any changes to the XPath expression you want to sign. The following extraction options are available:
    a. **Entire element** will sign the entire XPath
    b. **Content Only** will sign only the text content.
4. Click **Evaluate XPath** to verify your changes.
5. Click **OK**.

Emulation Options Settings

Configuring the emulation settings is only applicable if **WS-Security Mode** is enabled in the General Settings.

1. Choose your application server from the **Emulate** drop-down menu to automatically configure the emulation options.
2. Choose the appropriate version of your application server from the **Version** drop-down menu.

You can also configure the emulation options manually by choosing **Custom** from the **Emulate** drop-down menu. The following options will be available:

- **wsse URI**: Select the namespace URI of the WS-Security specification used.
- **wsu URI**: Select the utility namespace URI of the WS-Security specification used.
- **Qualify signed element ID attribute**: Select to qualify signed element ID attribute.
- **Qualify BinarySecurity Token attributes**: Select to qualify binary security token attributes with the wsse namespace.
- **Prefix BinarySecurity Token attribute values**: Select to prefix binary security token attributes with the wsse URI.

Input Type Tab

The **Input Type** tab is only available if the XML Signer tool is added as a standalone tool and not chained to another tool. The following options are available from the Input Type tab:

- **Text**: Use this option if you want to type or copy the XML document into the UI. Select the appropriate **MIME type**, enter the XML in the text field below the **Text** radio button.
- **File**: Use this option if you want to use an existing file. Click the Browse button to choose a file.
    - Check the **Persist as Relative Path** option if you want the path to this file to be saved as a path that is relative to the current configuration file. Enabling this option makes it easier to share tools across multiple machines. If this option is not enabled, the test or Responder will save the path to this file as an absolute path.

Usage Notes

You can use the XML Signer tool as a standalone tool at the tool level by right-clicking the main test suite node and selecting **Add New> Test** from the shortcut menu and then selecting **XML Signer** from the dialog that opens.You may also chain the XML Signer tool to a messaging tool by right-clicking the desired tool node and selecting **Add Output** from the shortcut menu and then selecting **XML Signer** from the dialog that opens. The messaging tool will use the transformed XML.

You can chain the XML Signer tool and the XML Encryption tool to a messaging tool to perform both encryption and XML signature on a SOAP message. For more information on the XML Encryption tool, see XML Encryption.

You can also chain any tool, such as an Edit or Browse tool, to the **XML Signer Tool** by right-clicking the desired **XML Signer Tool** node and selecting **Add Output** from the shortcut menu and then selecting **XML Signer** from the dialog that opens.

Unlimited Strength Java Cryptography Extension

In order to perform security operations using the XML Signature Verifier, XML Signer, or XML Encryption tools, or if using Key Stores, you will need to download and install the Unlimited Strength Java Cryptography Extension. For details, see JCE Prerequisite.

Related Tutorials

The following tutorial lesson demonstrates how to use this tool:

- WS-Security