Securing Web Service

Web Service security (WS-security) is an SOAP-based security standard that provides Web services with message-level integrity, confidentiality and authentication.

With WS-security, the Simple Object Access Protocol (SOAP) message contains a SOAP header, which includes signature, encryption information, protocols for processing the secured information, and security tokens for credential propagation.

A WS-Security policy file (WSSE) is associated with a Web service so that both inbound and outbound SOAP messages are handled according to the security policy in the WSSE file.

Geronimo 2.2 has two WS-security providers: Axis2 for Tomcat Web container and CXF for Jetty. They enable the following WS-security features in Web service development for Geronimo:

- **XML Security** - allows one to send along with the message a digital signature of it, which assures that no one modified the message content between the sender and receiver.
- **XML Encryption** -allows one to encrypt the message body or only its part using the given cryptography algorithm.
- **Username Tokens** - adds username and password values to the message header.
- **Security Assertions Markup Language (SAML) Tokens** - configured on web services via Geronimo deployment descriptors and/or annotations.
- **Timestamps** - specifies how long the security data remains valid.

In this guide, CXF/Jetty will be used as an example.

Enabling WS-security in Web service client

Configuring security properties

You can specify various properties using a <property> element in the <port> section in `geronimo-web.xml` for a CXF/Jetty client.
To configure ws-security properties, you only need to prefix each property with a `wss4j.in` file for inbound settings, or `wss4j.out` for outbound settings.
For example:

```
<property name="wss4j.out.action">UsernameToken Timestamp</property>
<property name="wss4j.out.user">foo</property>
<property name="wss4j.out.password">bar</property>
```

Enabling signed or encrypted SOAP messages

Geronimo allows the CXF/Jetty client to sendor receive the signed or encrypted SOAP messages. You can enable this feature inside <port> in the `geronimo-web.xml` at client side. The following code snippet is an example for both signing and encrypting:

```
<port>
<port-name>DoubleItPort</port-name>
<protocol>http</protocol>
<host>localhost</host>
<port>8080</port>
<uri>/doubleit/services/doubleit</uri>
...
<port>
```

Enabling WS-security at service side

Configuring Username token

Geronimo CXF/Jetty provides support for UsernameToken Profile at server side. For example, to involve UsernameToken profile for the Web service **CalculatorService**, add the following lines in `geronimo-web.xml`:

```
<servlet>
<servlet-name>CalculatorService</servlet-name>
<ws-security-binding>
<security-realm-name>geronimo-admin</security-realm-name>
<property name="wss4j.in.action">UsernameToken</property>
</ws-security-binding>
</servlet>
```

Enabling signed or encrypted SOAP messages

Similarly, you can enable the service side to send or receive signed or encrypted SOAP messages by configuring the <port> section in `geronimo-web.xml`. The following example passes security properties:

```
<<servlet>
<servlet-name>DoubleItServiceImpl</servlet-name>
<ws-security-binding>
<security-realm-name></security-realm-name>
<property name="wss4j.in.action">Signature Encrypt Timestamp</property>
<property name="wss4j.in.user">myservicekey</property>
<property name="wss4j.in.keyPassword">skpass</property>
<property name="wss4j.in.signaturePropFile">serviceKeystore.properties</property>
<property name="wss4j.in.decryptionPropFile">serviceKeystore.properties</property>

<property name="wss4j.out.action">Signature Encrypt Timestamp</property>
<property name="wss4j.out.user">myservicekey</property>
<property name="wss4j.out.signaturePropFile">serviceKeystore.properties</property>
<property name="wss4j.out.encryptionPropFile">serviceKeystore.properties</property>
<property name="wss4j.out.encryptionUser">myclientkey</property>
<property name="wss4j.out.signatureKeyIdentifier">DirectReference</property>
<property name="wss4j.out.keyPassword">skpass</property>
<property name="wss4j.out.encryptionSymAlgorithm">http://www.w3.org/2001/04/xmlenc#tripledes-cbc</property>
</ws-security-binding>
</servlet>
```