Authentication, Encryption, and Access Control

This topic explains how SOAtest assists with runtime security policy validation by enabling execution of complex authentication, encryption, and access control test scenarios.

Sections include:

- Overview
- Tutorial
- Related Topics
- Testing Oracle/BEA WebLogic Services with WS-Security
- JCE Prerequisite

Overview

To help you ensure that your security measures work flawlessly, SOAtest contains a vast array of security tools and options that help you construct and execute complex authentication, encryption, and access control test scenarios. For example:

- **XML Encryption tool:** The XML Encryption tool allows you to encrypt and decrypt entire messages or parts of messages using Triple DES, AES 128, AES 192, or AES 256. In WS-Security mode, Binary Security Tokens, X509IssuerSerial, and Key Identifiers are supported.
- **XML Signer tool:** The XML Signer tool allows you to digitally sign an entire message or parts of a message depending on your specific needs. In some cases it may be important to digitally sign parts of a document while encrypting other parts.
- **XML Signature Verifier tool:** The XML Signature Verifier tool allows for the verification of digitally signed documents using a public/private key pair stored within a key store file.
- **Key Stores:** The use of key stores in SOAtest allows you to encrypt/decrypt and digitally sign documents using public/private key pairs stored in a key store. Key stores in JKS, PEM, PKCS12, BKS, and UBER format can be used.
- **Username Tokens, SAML Tokens, X509 Tokens, or Custom Headers:** SOAtest supports sending custom SOAP Headers and includes templates for Username Tokens and SAML tokens.

Tutorial

For a step-by-step demonstration of how to apply SOAtest for validating authentication, encryption, and access control, see WS-Security. This tutorial covers encryption/decryption, digital signature, and the addition of SOAP Headers.

Related Topics

For more details on how to use SOAtest's tools to support your specific authentication, encryption, and access control validation needs, see the following sections.

Topic	Reference
WS-Security Policy	SOA Quality Governance and Policy Enforcement Adding Global Test Suite Properties
Custom Headers	Adding SOAP Headers Adding SAML Headers Adding Global Test Suite Properties
Tools	XML Encryption XML Signer XML Signature Verifier
General Security Settings (Authentication, Keystores, etc.)	Security Settings HTTP 1.0 HTTP 1.1 Adding Global Test Suite Properties
HTTPS and SSL	Configuring for Services Deployed Over HTTPS
SAML	Adding SAML Headers SAML 1.1 Assertion Options SAML 2.0 Assertion Options

Testing Oracle/BEA WebLogic Services with WS-Security

If your services are configured with WS-Security XML security policies, then you can configure SOAtest with the necessary settings in order to interoperate with WebLogic.

To help you configure these settings, a sample SOAtest project WebLogicWSS.tst is included under `[SOAtest install dir]/examples/tests`. WebLogicWSS.tst is not an executable test; it intended to serve as a reference, allowing you to compare a working configuration that has been verified by Parasoft against your own. This example configuration has been tested to work with WebLogic 9.2 and later.

This example assumes that default sign, encrypt and UsernameToken (ut) policies are being used by your WebLogic application. It also assumes that the wss_client certificate (the client public key) has been imported to WebLogic's DemoTrust keystore.

Please note how:

- The signature and encryption tests in WebLogicWSS.tst include a WS-Security header with an X509 token configured.
- The various encryption and signature tools are setup for various WS-Security scenarios.
- Two certificate alias are used by the various operations.

If you are using the default policies or policies that are built off of the defaults, configure your test settings to match this example in terms of the options selected.

For more information about WebLogic security policies, please refer to Oracle's e-docs sites.

JCE Prerequisite

See JCE Prerequisite.