Security Testing - Introduction

This topic provides an overview of how SOAtest facilitates two types of security testing:

- Authentication, encryption, and access control (i.e., runtime security policy validation).
- Hybrid security analysis, which integrates penetration testing with runtime error detection.

About Authentication, Encryption, and Access Control

SOAtest assists with runtime security policy validation by enabling execution of complex authentication, encryption, and access control test scenarios. SOAtest includes security support for testing services with security layers.

At the transport level, SOAtest supports SSL (both server and client authentication), basic, Digest and Kerberos authentication.

At the message level, SOAtest supports WS-Security including X509, SAML, Username security tokens, XML Encryption and XML Digital Signature. It allows for security token validation as well as negative tests that ensure proper enforcement of message integrity and authentication.

> **Learning More**
>
> For details on how to perform this validation, see Authentication, Encryption, and Access Control.

About Hybrid Security Analysis

SOAtest's hybrid security analysis takes the functional tests that you and your team have already defined and uses them to perform a fully-automated assessment of where security attacks actually penetrate the application.

This hybrid analysis:

1. Uses penetration testing to automatically generate and run penetration attack scenarios against your existing web or service functional test scenarios.
2. Uses runtime error detection to monitor the back-end of the application during test execution in order to determine whether security is actually compromised (and whether other runtime errors occur as well).
3. Correlates each reported runtime error with the functional test that was being run when the error was detected—allowing you to trace each reported error to a the specific use case related to the problem.

The two key components of hybrid analysis—penetration testing and runtime error detection—can also be used independently of one another.

Penetration Testing

SOAtest's penetration testing can generate and run a variety of attack scenarios (such as Parameter Fuzzing, SQL and XPath injections, Cross Site Scripting, XML Bombs, and more) against your functional test suites.

If you are not able or ready to configure your application server for runtime error detection, you can still use penetration testing to generate and run attack scenarios, then use alternative strategies to determine if the attacks caused security breaches.

Runtime Error Detection

SOAtest's runtime error detection monitors the application from the back-end as SOAtest tests executes and alerts you if security breaches or other runtime defects (such as race conditions, exceptions, resource leaks) actually occur.

You may want to perform runtime error detection both with and without penetration testing. This way, you can ensure that error detection covers both the exact use case functionality captured in your test cases as well as the simulated attacks based on this functionality.

> **Learning More**
>
> For details on how to perform these analyses, see Penetration Testing and Performing Runtime Error Detection.