Conformance Program Specification for the OASIS Security Assertion Markup Language (SAML)

Document identifier: draft-sstc-conform-spec-08

Location: http://www.oasis-open.org/committees/security/docs

Publication date: 10 January 2002

Maturity Level: Committee Working Draft

Send comments to: security-services-comment@lists.oasis-open.org *unless* you are subscribed to the security-services list for committee members -- send comments there if so. Note: Before sending messages to the security-services-comment list, you must first subscribe. To subscribe, send an email message to security-services-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

Contributors:

Marc Chanliau, Netegrity
Robert Griffin, Entrust (editor)
Hal Lockhart, Entegrity
Eve Maler, Sun Microsystems
Prateek Mishra, Netegrity
Mike Myers
Charles Norwood, SAIC
Mark O'Neill, Vordel
Tony Palmer, Vordel
Lynne Rosenthal, NIST
Krishna Sankar, Cisco Systems
Mark Skall, NIST

Rev	What
001	Initial version
002	Strawman profiles, test cases and process
003	Revisions from 1-June-2001 review; added example of test case
004	Revisions from 18-June-2001 review; modified to reflect conformance clause
005	Additions to test cases
006	Additions to test cases; HTTP profile mandatory
007	Includes conformance clause; SOAP binding mandatory
007a	Draft using assertions rather partitions as basis of conformance
007b	Draft using bindings rather than partitions as basis of conformance

007c	Stylistic edits and added OASIS notices to 007a
008	Revised, using bindings approach, to correct references, include issue

CONFORMANCE PROGRAM FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) 1

1 INTRODUCTION 4

1.1 SCOPE OF THE CONFORMANCE PROGRAM 4

1.2 NOTATION 4

2 CONFORMANCE CLAUSE 5

2.1 SPECIFICATION OF THE SAML STANDARD 5

2.2 DECLARATION OF SAML CONFORMANCE 5

2.3 MANDATORY/OPTIONAL ELEMENTS IN SAML CONFORMANCE 6

2.4 IMPACT OF EXTENSIONS ON SAML CONFORMANCE 7

3 CONFORMANCE PROCESS 8

3.1 IMPLEMENTATION AND APPLICATION CONFORMANCE 8

3.2 PROCESS FOR DECLARING CONFORMANCE 9

4 TECHNICAL REQUIREMENTS FOR SAML CONFORMANCE 10

4.1 TEST GROUP 1 – SOAP OVER HTTP PROTOCOL BINDING 10

4.1.1 Test Case 1-1: SOAP Protocol Binding: Valid Authentication Assertion Received in Valid Response to Valid Authentication Query. 10

4.1.2 Test Case 1-2: SOAP Protocol Binding: Valid Authentication Assertion Artefact Returned in Valid Response to Valid Authentication Query. 10

4.1.3 Test Case 1-3: SOAP Protocol Binding: Valid Authentication Assertion Returned in Valid Response to Valid Authentication Query with artefact. 11

4.1.4 Test Case 1-4: SOAP Protocol Binding: Valid Attribute Assertion Received in Valid Response to Valid Attribute Query. 11

4.1.5 Test Case 1-5: SOAP Protocol Binding: Valid Attribute Assertion Artefact Returned in Valid Response to Valid Attribute Query. 12

4.1.6 Test Case 1-6: SOAP Protocol Binding: Valid Authentication Assertion Returned in Valid Response to Valid Attribute Query. 12

4.1.7 Test Case 1-7: SOAP Protocol Binding: Valid Authorization Decision Assertion Received in Valid Response to Valid Authorization Decision Query. 13

4.1.8 Test Case 1-8: SOAP Protocol Binding: Valid Authorization Decision Assertion Artefact Returned in Valid Response to Valid Authorization Decision Query. 13

4.1.9 Test Case 1-9: SOAP Protocol Binding: Valid Authorization Decision Assertion Returned in Valid Response to Valid Authentication Query. 14

4.2 TEST GROUP 2: SOAP PROFILE 14

4.2.1 Test Case 2-1: SOAP Profile: Valid Authentication Assertion Received in Valid Response to Valid Authentication Query. 14

4.2.2 Test Case 2-2: SOAP Profile: Valid Authentication Assertion Artefact Returned in Valid Response to Valid Authentication Query. 15

4.2.3 Test Case 2-3: SOAP Profile: Valid Authentication Assertion Returned in Valid Response to Valid Authentication Query with artefact. 15

4.2.4 Test Case 2-4: SOAP Profile: Valid Attribute Assertion Received in Valid Response to Valid Attribute Query. 15

4.2.5 Test Case 2-5: SOAP Profile: Valid Attribute Assertion Artefact Returned in Valid Response to Valid Attribute Query. 16

4.2.6 Test Case 2-6: SOAP Profile: Valid Authentication Assertion Returned in Valid Response to Valid Attribute Query. 16

4.2.7 Test Case 2-7: SOAP Profile: Valid Authorization Decision Assertion Received in Valid Response to Valid Authorization Decision Query. 16

4.2.8 Test Case 2-8: SOAP Profile: Valid Authorization Decision Assertion Artefact Returned in Valid Response to Valid Authorization Decision Query. 17

4.2.9 Test Case 2-9: SOAP Profile: Valid Authorization Decision Assertion Returned in Valid Response to Valid Authentication Query. 17

4.3 TEST GROUP 3 – WEB BROWSER PROFILES 17

4.3.1 Test Case 3-1: HTTP Web Browser/Artefact Profile: Valid Authentication Assertion Artefact Produced in Response to Valid Authentication Query. 17

4.3.2 Test Case 3-2: HTTP Web Browser/Artefact Profile: Valid Authentication Assertion Produced in Response to Valid Authentication Query with Artefact. 18

4.3.3 Test Case 3-3: Web Browser/Post Profile: Valid Authentication Assertion Produced in Response to Valid Authentication Query. 18

5 TEST SUITE 19

5.1 REFERENCE ARCHITECTURE ..... **ERROR! BOOKMARK NOT DEFINED.**

5.2 INFRASTRUCTURE ..... **ERROR! BOOKMARK NOT DEFINED.**

5.3 USING THE TEST SUITE ..... **ERROR! BOOKMARK NOT DEFINED.**

5.4 TEST RESULT TABULATION AND REPORTING ..... **ERROR! BOOKMARK NOT DEFINED.**

6 CONFORMANCE SERVICES 20

6.1 ESTABLISHING A SAML TESTING SERVICE ..... **ERROR! BOOKMARK NOT DEFINED.**

7 REFERENCES 21

APPENDIX A. NOTICES 22

1 Introduction

This document describes the program and technical requirements for the SAML conformance system.

1.1 Scope of the Conformance Program

SAML deals with a rich set of functionalities ranging from authentication assertions to assertions for policy enforcement. Not all software might choose to implement all the SAML specifications. In order to achieve compatibility and interoperability, applications and software need to be certified for conformance in a uniform manner. The SAML conformance effort aims at fulfilling this need.

The deliverables of the SAML conformance effort include:

- Conformance Clause, defining at a high-level what conformance means for the SAML standard
- Conformance Program specification, defining how an implementation or application establishes conformance
- Conformance Test Suite. This is a set of test programs, result files and report generation tools that can be used by vendors of SAML-compliant software, buyers interested in confirming SAML compliance of software, and testing labs running conformance tests on behalf of vendors or buyers.

Section 2 of this document provides the SAML Conformance Clause. Section 3 deals with defining and specifying the process by which conformance to the SAML specification can be demonstrated and certified. Section 4 elaborates the technical requirements which constitute conformance; this includes both the levels of conformance that may be demonstrated and the requirements for each of those levels of conformance. Section 5 describes the test suite for SAML, including the processes for using the test suite to establish conformance, and the policies and procedures relating to those processes. Section 6 defines the services which are available to assist in establishing conformance.

1.2 Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119 **[NIST/ITL]** “What is this thing called conformance” [Rosenthal, Brady; NIST/ITL Bulletin,January 2001] http://www.itl.nist.gov/div897/ctg/conformance/bulletin-conformance.htm.

[RFC2119].

2 Conformance Clause

The objectives of the SAML Conformance Clause are to:

1. Ensure a common understanding of conformance and what is required to claim conformance
2. Promote interoperability in the exchange of authentication and authorization information
3. Promote uniformity in the development of conformance tests

The SAML Conformance Clause specifies explicitly all the requirements that have to be satisfied to claim conformance to the SAML standard.

2.1 Specification of the SAML Standard

The following four specifications, in addition to this SAML conformance program specification, comprise the proposed Version 1.0 specification for the SAML standard:

- Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) **[SAMLCore]**
- Security Considerations for the OASIS Security Assertion Markup Language (SAML) **[SAMLSec]**
- Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) **[SAMLBind]**
- Glossary for the OASIS Security Assertion Markup Language (SAML) **[SAMLGloss]**

Although additional documents might use or reference the SAML standard (such as whitepapers, descriptions of custom profiles, and position papers referencing particular issues), they do not constitute part of the standard.

2.2 Declaration of SAML Conformance

Conformance to the SAML standard may be declared for the entire standard or for a subset of the standard, based on the requirements that a given implementation or application claims to meet. That is, requirements can be applied at varying levels, so that a given implementation or application of the SAML standard can achieve clearly defined conformance with all or part of the entire set of requirements.

SAML conformance must be expressed in terms of which SAML bindings are supported by a given application or implementation. The application or implementation claiming conformance to the SAML standard must support the SOAP protocol binding, at least with reqard to required elements of the binding; the application or implementation does not have to support optional elements of the binding, but it must state whether or not the optional elements are supported. It must also be able to detect and handle optional elements in messages and/or assertions that it receives from another SAML implementation applicaiton.

An application or implementation may also support the web browser profiles and/or the SOAP profile.

For any binding for which an application or implementation claims conformance, the level of conformance must then be specified in each of these dimensions:

- Whether the application or implementation acts as requestor or responder or both requestor and responder of the SAML messages in the supported bindings and profiles.
- Which assertions the application or implementation supports for each supported binding.

Table 1 shows the protocols, protocol bindings, and profiles applicable to each SAML assertion. For each SAML assertion to which an application or implementation claims conformance, the claim must stipulate which of the cells under Protocol, Protocol Binding, and Profile are supported.

165 **Table 1: Protocols, Protocol Bindings and Profiles for SAML Assertions**

Binding	Producer / Consumer	Relevant Asertions
SOAP over HTTP protocol binding (required)	Consumer (uses AuthenticationQuery to request assertion)	Authentication Assertion, Attribute Assertion and/or Authentication Decision Assertion
	Producer: (uses AuthenticationResponse to return assertion)	Authentication Assertion, Attribute Assertion and/or Authentication Decision Assertion
SOAP Profile (optional)	Consumer (requests assertion)	Authentication Assertion, Attribute Assertion and/or Authentication Decision Assertion
	Producer (returns assertion)	Authentication Assertion, Attribute Assertion and/or Authentication Decision Assertion
Browser/Artefact Profile (optional)	Consumer (requests assertion)	Authentication Assertion
	Consumer (returns assertion)	Authentication Assertion
Browser/POST Profile (optional)	Consumer (requests assertion)	Authentication Assertion
	Producer (returns assertion)	Authentication Assertion

166

167 An application or implementation should express its level of conformance in terminology such as the
168 following:

169 *[Application or implementation] as both consumer and producer supports all SAML protocol bindings and*
170 *profiles, for all assertions and required elements. No optional elements for the bindings and profiles are*
171 *supported.*

172 *[Application or implementation] as both consumer and producer supports the SOAP protocol binding for all*
173 *assertions and required elements. It also supports the Conditions optional elements for all assertions in the*
174 *SOAP protocol binding. It does not support the Web Browser Profile and the SOAP profile for any*
175 *assertion.*

176 *[Application or implementation] as both consumer and producer supports the SOAP protocol binding for all*
177 *assertions, for all assertions and required elements. It also support the Web Browser Profile for*
178 *Authentication Assertion and all required elements. No optional elements for the bindings and profiles are*
179 *supported.*

180 An application or implementation that claims conformance for a particular binding or profile must support all
181 required elements of that binding or profile. It must also state which assertions are supported and which, if
182 any optional elements for that binding are supported.

183 2.3 Mandatory/Optional Elements in SAML Conformance

184 The SOAP protocol binding must be implemented by all implementations or applications claiming SAML
185 conformance, for all assertions claimed as supported through a binding a profile. (see Appendix B: Issues)

An application or implementation claiming conformance for a binding and/or profile must include all elements that are specified as mandatory in the SAML documents. For each of the bindings and profiles, there are also optional elements that an application or implementation is not required to implement. However, the implementation or application must be able to handle (in most cases, reject) assertions or messages containing optional elements that it does not understand.

For example, the SOAP profile stipulates that “every assertion MUST be signed by the issuer”. That is, digital signature is required on the assertion. However, a server-side certificate is required with SSLv3 or TLS1.0 only if message confidentiality is being claimed for the SAML implementation or application, above and beyond the required functionality.

The test cases for SAML conformance are intended to check for support of mandatory requirements. They also check whether an implementation or application accepts and properly handles optional assertion elements (such as CONDITION) who value the implementation or application does not recognize. The test suite does not check for handling of implementation- or application-specific values for optional elements.

2.4 Impact of Extensions on SAML Conformance

SAML supports extensions to assertions, protocols, protocol bindings and profiles. An application or implementation may claim conformance to SAML only if its extensions (if any) meet the following requirements:

- Extensions shall not re-define semantics for existing functions.
- Extensions shall not alter the specified behavior of interfaces defined in this standard.
- Extensions may add additional behaviors.
- Extensions shall not cause standard-conforming functions (i.e., functions that do not use the extensions) to execute incorrectly.

SAML bindings and profils can be extended so long as the above conditions are met. It is requested that, if a system is extending the SAML assertions:

- The mechanism for determining application conformance and the extensions shall be clearly described in the documentation, and the extensions shall be marked as such;
- Extensions shall follow the spirit, principles and guidelines of the SAML specification, that is, the specifications must be extended in a standard manner as defined in the extension fields.
- In the case where an implementation has added additional behaviors, the implementation shall provide a mechanism whereby a conforming application shall be recognized as such, and be executed in an environment that supports the functional behavior defined in this standard

Extensions are outside the scope of conformance. There are no mechanisms specified to validate and verify the extensions. This section contains the recommended guidelines for extensions.

3 Conformance Process

As discussed in the article “What is this thing called conformance” **[NIST/ITL]**, conformance can comprise any of several levels of formal process:

- **Conformance testing** (also called conformity assessment) is the execution of automated or non-automated scripts, processes or other mechanisms to determine whether an application or implementation of a specification deviates from that specification. For SAML, conformance testing means the running of (some or all) tests within the SAML Conformance Test Suite. Conformance testing performed by implementers early on in the development process can find and correct their errors before the software reaches the marketplace, without necessarily being part of either a validation or certification process.
- **Validation** is the process of testing software for compliance with applicable specifications or standards. The validation process consists of the steps necessary to perform the conformance testing by using an official test suite in a prescribed manner.
- **Certification** is the acknowledgment that a validation has been completed and the criteria established by the certifying organization for issuing a certificate have been met. Successful completion of certification results in the issuance of a certificate (or brand) indicating that the implementation conforms to the appropriate specification. It is important to note that certification cannot exist without validation, but validation can exist without certification.

The conformance process for SAML is based on validation rather than certification. That is, no certifying organization has been established with the responsible for issuing a statement of conformance with regard to an application or implementation. Therefore, an implementer who has validated SAML conformance by means of conformance testing may not legitimately use the term “certified for SAML conformance”. Until and if a certification process is in place, vendor declaration of validation will be the only means of assertintg that conformance testing has been performed.

The conformance process does not stipulate whether validation is performed by the implementor, by a third-party, or by the customer of an application or implementation. Rather, the conformance process describes the way in which conformance testing should be done in order to demonstrate that an application or implementation correctly performs the functionality specified in the standard. Validation achieved through the SAML conformance process provides software developers and users assurance and confidence that the product behaves as expected, performs functions in a known manner, and possesses the prescribed interface or format.

The SAML Technical Committee is responsible for generating the materials that allow vendors, customers, and third parties to evaluate software for SAML conformance. These materials include:

- Documentation describing test cases, linked to use cases and requirements
- Test suite, based on those test cases, that can be run against an implementation to demonstrate any of the several levels/profiles of conformance defined in the conformance clause of the SAML specification
- Documentation describing how to run the test suite, interpret the results, and resolve disputes regarding the results of the tests

The SAML Technical Committee is not, however, responsible for testing of particular implementations.

3.1 Implementation and Application Conformance

SAML Conformance is applicable to:

- Implementations of SAML assertions, protocols and bindings. These could be in the form of toolkits, products incorporating SAML components, or reference implementations that demonstrate the use of SAML components.

- Applications that produce or consume SAML protocol bindings or that execute on SAML implementations (for example, using a SAML toolkit to support multi-domain single-signon)

A conforming **implementation** shall meet all the following criteria:

4. The implementation shall support all the required interfaces defined within this standard for a given binding or profile. It shall also specify which assertions relevant to that binding or profile are supported. The implementation shall support the functional behavior described in the standard.
5. An implementation may provide additional or enhanced features or functionality not required by the SAML Specification. These non-standard extensions shall not alter the specified behavior of interfaces or functionality defined in the specification.
6. The implementation may provide additional or enhanced facilities not required by this standard. These non-standard extensions shall not alter the specified behavior of interfaces defined in this standard. They may add additional behaviors. In these circumstances, the implementation shall provide a mechanism whereby a SAML conforming application shall be recognized as such, and be executed in an environment that supports the functional behavior defined in this standard.

A conforming **application** shall meet all the following criteria:

1. The application shall be able to execute on any conforming implementation.
2. If an application requires a particular feature set that is not available on a specific implementation, then the application must act within the bounds of the SAML specification even though that means that the application may not perform any useful function. Specifically, the application shall do no harm, and shall correctly return resources and vacate memory upon discovery that a required element is not present.

3.2 Process for Declaring Conformance

The following process should be followed in declaring that an application or implementation conforms to the SAML standard:

1. Determine which bindings and protocols will be asserted as conforming.
2. Obtain the test suite for the SAML standard from [tbs]
3. Validate the application or implementation by execute those conformance tests from the test suite which are relevant to the conformance being asserted.
4. Send the statement claiming conformance to the Security Services Technical Committee at [tbs] so that it can be posted on the SAML web site. A statement of any bindings and profiles which are being used that are not part of the SAML standard should also be sent to the Security Services Technical Committee at the same time for posting on the SAML web site.

4 Technical Requirements for SAML Conformance

This section defines the tecnical criteria which apply to declaring conformance to the SAML standard. The requirements are specified as test cases.

Each test case includes:

- A description of the test purpose (that is, what is being tested – the conditions, requirements, or capabilities which are to be addressed by a particular test)
- The pass/fail criteria
- A reference to the requirement in the requirements document **[SAMLReqs]** relevant to the test case
- A reference to the section in the standard from which the test case is derived (that is, traceability back to the specification)

For each assertion, both required tests for producing and consuming the assertion, as well as tests related to protocols, bindings and profiles are specified.

4.1 Test Group 1 – SOAP over HTTP Protocol Binding

The test cases in this test group check for conformance to SOAP Protocol Binding for the SAML standard. Any implementation or application claiming conformance to SAML must be able to execute these test cases successfully, even if that support is incidental to the primary purposes of the application or implementation.

4.1.1 Test Case 1-1: SOAP Protocol Binding: Valid Authentication Assertion Received in Valid Response to Valid Authentication Query.

Description: This test case requests and receives an authentication assertion created by an iimplementation-under-test using the AuthenticationRequest protocol in the SOAP binding. It then confirms that the authentication assertion returned by the implementation-under-test is valid for all required functionality.

Pass/Fail Criteria: Authentication assertion contains all required elements in the right format and sequence, AuthenticationQuery is accepted by implementation-under-test, and AuthenticationResponse contains all required elements in correct sequence.

Requirements Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, sections 2.4.3 and 3

draft-sstc-bindings-model-09, section 3.1.

Implementation notes: Test program implementing this test case uses the SOAP over HTTP binding of the AuthenticationQuery and AuthenticationResponse protocols to obtain the Authentication Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.1.2 Test Case 1-2: SOAP Protocol Binding: Valid Authentication Assertion Artefact Returned in Valid Response to Valid Authentication Query.

Description: This test case submits a SOAP message containing authentication credentials to an implementation-under-test, requesting an authentication artefact. It checks that the implementation-under-test returns a valid authentication assertion artefact in a valid AuthenticationResponse. It then submit the artefact to the application/implementation-under-test. Finally, it checks that the returned authentication assertion is valid.

Pass/Fail Criteria: Authentication assertion artefact returned by implementation-under-test must be contain all required information in the right sequence and format. Any optional information included (including conditions) must not compromise the validity of the required information.

Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, sections 2.4.3 and 3

draft-sstc-bindings-model-09, section 3.1.

Implementation notes: Test program implementing this test case establishes successful execution of the test case by inspection of the format of the returned assertion artefact.

4.1.3 Test Case 1-3: SOAP Protocol Binding: Valid Authentication Assertion Returned in Valid Response to Valid Authentication Query with artefact.

Description: This test case requests and receives an authentication assertion artefact created by an implementation-under-test using the AuthenticationRequest protocol in the SOAP binding. It then confirms that the returned authentication assertion is valid for all required functionality.

Pass/Fail Criteria: Authentication assertion contains all required elements in the right format and sequence, AuthenticationQuery is accepted by implementation-under-test, and AuthenticationResponse contains all required elements in correct sequence.

Requirements Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, sections 2.4.3 and 3

draft-sstc-bindings-model-09, section 3.1

Implementation notes: Test program implementing this test case uses the SOAP over HTTP binding of the AuthenticationQuery and AuthenticationResponse protocols to obtain the Authentication Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.1.4 Test Case 1-4: SOAP Protocol Binding: Valid Authentication Assertion Query Received

Description: This test case receives an authentication assertion query created by an implementation-under-test using the AuthenticationRequest protocol in the SOAP binding. It then confirms that the returned authentication query is valid for all required functionality.

Pass/Fail Criteria: AuthenticationQuery contains all required elements in the right format and sequence.

Requirements Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, sections 2.4.3 and 3

draft-sstc-bindings-model-09, section 3.1

Implementation notes: Test program implementing this test case uses the SOAP over HTTP binding of the AuthenticationQuery and AuthenticationResponse protocols to obtain the Authentication Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.1.5 Test Case 1-5: SOAP Protocol Binding: Valid Attribute Assertion Received in Valid Response to Valid Attribute Query.

Description: This test case requests and receives an attribute assertion created by an iimplementation-under-test using the AttributeRequest protocol in the SOAP binding. It then confirms that the attribute assertion returned by the implementation-under-test is valid for all required functionality.

Pass/Fail Criteria: Auttribute assertion contains all required elements in the right format and sequence, AttributeQuery is accepted by implementation-under-test, and AttributeResponse contains all required elements in correct sequence.

Requirements Reference: **R-AUTHZ**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Sectiosn 2.4.5 and 3

draft-sstc-bindings-model-09, section 3.1.

Implementation notes: Test program implementing this test case uses the SOAP over HTTP bindings of the AttributeQuery and AttributeResponse protocols to obtain the Attribute Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.1.6 Test Case 1-6: SOAP Protocol Binding: Valid Attribute Assertion Artefact Returned in Valid Response to Valid Attribute Query.

Description: This test case submits a SOAP message containing attribute credentials to an implementation-under-test, requesting an attribute artefact. It checks that the implementation-under-test returns a valid attribute assertion artefact in a valid AttributeResponse. It then submit the artefact to the application/implementation-under-test. Finally, it checks that the returned attribute assertion is valid.

Pass/Fail Criteria: Attribute assertion artefact returned by implementation-under-test must be contain all required information in the right sequence and format. Any optional information included (including conditions) must not compromise the validity of the required information.

Reference: **R-AUTHZ**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Sections 2.4.5 and 3

draft-sstc-bindings-model-09, section 3.1.

Implementation notes: Test program implementing this test case establishes successful execution of the test case by inspection of the format of the returned assertion artefact.

4.1.7 Test Case 1-7: SOAP Protocol Binding: Valid Attribute Assertion Returned in Valid Response to Valid Attribute Query.

Description: This test case requests and receives an attribute assertion created by an implementation-under-test using the AttributeRequest protocol in the SOAP binding. It then confirms that the attribute assertion is valid for all required functionality.

Pass/Fail Criteria: Attribute assertion contains all required elements in the right format and sequence, AttributeQuery is accepted by implementation-under-test, and AttributeResponse contains all required elements in correct sequence.

Requirements Reference: **R-AUTHZ**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, SSections 2.4.5 and 3

draft-sstc-bindings-model-09, section 3.1

Implementation notes: Test program implementing this test case uses the SOAP over HTTP binding of the AttributeQuery and AttributeResponse protocols to obtain the Auttribute Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.1.8 Test Case 1-8: SOAP Protocol Binding: Valid Attribute Query Received

Description: This test case receives an attribute assertion query created by an implementation-under-test using the AttributeRequest protocol in the SOAP binding. It then confirms that the returned authentication query is valid for all required functionality.

Pass/Fail Criteria: AuthenticationQuery contains all required elements in the right format and sequence.

Requirements Reference: **R-AUTHZ**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, sections 2.4.5 and 3

draft-sstc-bindings-model-09, section 3.1

Implementation notes: Test program implementing this test case uses the SOAP over HTTP binding of the AttributeQuery and Response protocols to obtain the Attribute Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.1.9 Test Case 1-9: SOAP Protocol Binding: Valid Authorization Decision Assertion Received in Valid Response to Valid Authorization Decision Query.

Description: This test case requests and receives an authentication assertion created by an iimplementation-under-test using the AuthenticationRequest protocol in the SOAP binding. It then confirms that the authentication assertion returned by the implementation-under-test is valid for all required functionality.

Pass/Fail Criteria: Authorization decision assertion contains all required elements in the right format and sequence, AuthorizationQuery is accepted by implementation-under-test, and AuthorizationResponse contains all required elements in correct sequence.

Requirements Reference: **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.4 and 3

draft-sstc-bindings-model-09, section 3.1.

Implementation notes: Test program implementing this test case uses the SOAP over HTTP bindings of the AuthorizationQuery and AuthorizationResponse protocols to obtain the Authorization decision Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.1.10 Test Case 1-10: SOAP Protocol Binding: Valid Authorization Decision Assertion Artefact Returned in Valid Response to Valid Authorization Decision Query.

Description: This test case submits a SOAP message containing an authorization decision request to an implementation-under-test, requesting an authorization decision artefact. It checks that the implementation-under-test returns a valid authorization decision assertion artefact in a valid AuthorizationResponse. It then submit the artefact to the application/implementation-under-test. Finally, it checks that the returned authorization decision assertion is valid.

Pass/Fail Criteria: Authorization decision assertion artefact returned by implementation-under-test must be contain all required information in the right sequence and format. Any optional information included (including conditions) must not compromise the validity of the required information.

Reference: **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Sectios 2.4.4 and 3

draft-sstc-bindings-model-09, section 3.1.

Implementation notes: Test program implementing this test case establishes successful execution of the test case by inspection of the format of the returned assertion artefact.

4.1.11 Test Case 1-11: SOAP Protocol Binding: Valid Authorization Decision Assertion Returned in Valid Response to Valid Query.

Description: This test case requests and receives an authorization decision assertion created by an implementation-under-test using the AuthorizationRequest protocol in the SOAP over HTTP binding. It then confirms that the uthorization decision assertion is valid for all required functionality.

Pass/Fail Criteria: Authorization decision assertion contains all required elements in the right format and sequence, AuthorizzationQuery is accepted by implementation-under-test, and AuthorizationResponse contains all required elements in correct sequence.

Requirements Reference: **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Sections 2.4.4 and 3

draft-sstc-bindings-model-09, section 3.1

Implementation notes: Test program implementing this test case uses the SOAP over HTTP protocol bindings of the AuthorizationQuery and AuthorizationResponse protocols to obtain the Authorization decision Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.1.12 Test Case 1-12: SOAP Protocol Binding: Valid Authorization Decision Assertion Query Received

Description: This test case receives an authorization decision assertion query created by an implementation-under-test using the AuthorizationRequest protocol in the SOAP binding. It then confirms that the received query is valid for all required functionality.

Pass/Fail Criteria: AuthorizationQuery contains all required elements in the right format and sequence.

Requirements Reference: **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, sections 2.4.4 and 3

draft-sstc-bindings-model-09, section 3.1

Implementation notes: Test program implementing this test case uses the SOAP over HTTP binding of the AuthenticationQuery and AuthenticationResponse protocols to obtain the Authentication Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.2 Test Group 2: SOAP Profile

The test cases in this test group check for conformance to the SOAP Profile for the SAML standard. upport of the SOAP Profile is optional. Any implementation or application claiming conformance to the SOAP Profile of SAML must be able to execute these test cases successfully.

4.2.1 Test Case 2-1: SOAP Profile: Valid Authentication Assertion Received in Valid Response to Valid Authentication Query.

Description: This test case uses the SOAP profile to request and receive an authentication assertion created by an implementation-under-test. It then confirms that the authentication assertion returned by the implementation-under-test is valid for all required functionality.

Pass/Fail Criteria: Authentication assertion contains all required elements in the right format and sequence,

Requirements Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4..3

draft-sstc-bindings-model-09, section 4.2.

Implementation notes: Test program implementing this test case uses the SOAP profile to obtain the Authentication Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.2.2 Test Case 2-2: SOAP Profile: Valid Authentication Assertion Artefact Returned in Valid Response to Valid Authentication Query.

Description: This test case submits a SOAP message containing authentication credentials to an implementation-under-test, requesting an authentication artefact. It checks that the implementation-under-test returns a valid authentication assertion artefact. It then submit the artefact to the application/implementation-under-test. Finally, it checks that the returned authentication assertion is valid.

Pass/Fail Criteria: Authentication assertion artefact returned by implementation-under-test must be contain all required information in the right sequence and format. Any optional information included (including conditions) must not compromise the validity of the required information.

Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.3

draft-sstc-bindings-model-09, section 4.2

Implementation notes: Test program implementing this test case establishes successful execution of the test case by inspection of the format of the returned assertion artefact.

4.2.3 Test Case 2-3: SOAP Profile: Valid Authentication Assertion Returned in Valid Response to Valid Authentication Query with artefact.

Description: This test case uses the SOAP profile to request and receive an authentication assertion artefact created by an implementation-under-test. It then confirms that the returned authentication assertion is valid for all required functionality.

Pass/Fail Criteria: Authentication assertion contains all required elements in the right format and sequence.

Requirements Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.3

draft-sstc-bindings-model-09, section 4.2

Implementation notes: Test program implementing this test case uses the SOAP profile to obtain the Authentication Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.2.4 Test Case 2-4: SOAP Profile: Valid Attribute Assertion Received in Valid Response to Valid Attribute Query.

Description: This test case uses the SOAP profile to request and receive an attribute assertion created by an iimplementation-under-test. It then confirms that the attribute assertion returned by the implementation-under-test is valid for all required functionality.

Pass/Fail Criteria: Auttribute assertion contains all required elements in the right format and sequence,

Requirements Reference: **R-AUTHZ**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.5

draft-sstc-bindings-model-09, section 4.2.

Implementation notes: Test program implementing this test case uses the SOAP profile to obtain the Attribute Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.2.5 Test Case 2-5: SOAP Profile: Valid Attribute Assertion Artefact Returned in Valid Response to Valid Attribute Query.

Description: This test case submits a SOAP message requesting an attribute artefact. It checks that the implementation-under-test returns a valid attribute assertion artefact. It then submits the artefact to the application/implementation-under-test. Finally, it checks that the returned attribute assertion is valid.

Pass/Fail Criteria: Attribute assertion artefact returned by implementation-under-test must be contain all required information in the right sequence and format. Any optional information included (including conditions) must not compromise the validity of the required information.

Reference: **R-AUTHZ**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.5

draft-sstc-bindings-model-09, section 4.2.

Implementation notes: Test program implementing this test case establishes successful execution of the test case by inspection of the format of the returned assertion artefact.

4.2.6 Test Case 2-6: SOAP Profile: Valid Attribute Assertion Returned in Valid Response to Valid Attribute Query.

Description: This test case uses the SOAP profile to request and receive an attribute assertion created by an implementation-under-test. It then confirms that the attribute assertion is valid for all required functionality.

Pass/Fail Criteria: Attribute assertion contains all required elements in the right format and sequence,

Requirements Reference: **R-AUTHZ**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.5

draft-sstc-bindings-model-09, section 4.2

Implementation notes: Test program implementing this test case uses the SOAP profile to obtain the Auttribute Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.2.7 Test Case 2-7: SOAP Profile: Valid Authorization Decision Assertion Received in Valid Response to Valid Authorization Decision Query.

Description: This test case uses the SOAP Profile to request and receive an authorization decision assertion created by an iimplementation-under-test. It then confirms that the authorization decision assertion returned by the implementation-under-test is valid for all required functionality.

Pass/Fail Criteria: Authorization decision assertion contains all required elements in the right format and sequence.

Requirements Reference: **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.4

draft-sstc-bindings-model-09, section 4.2.

Implementation notes: Test program implementing this test case uses the SOAP profile to obtain the Authorization decision Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.2.8 Test Case 2-8: SOAP Profile: Valid Authorization Decision Assertion Artefact Returned in Valid Response to Valid Authorization Decision Query.

Description: This test case submits a SOAP message containing an authorization decision request to an implementation-under-test, requesting an authorization decision artefact. It checks that the implementation-under-test returns a valid authorization decision assertion artefact. It then submit the artefact to the application/implementation-under-test. Finally, it checks that the returned authorization decision assertion is valid.

Pass/Fail Criteria: Authorization decision assertion artefact returned by implementation-under-test must be contain all required information in the right sequence and format. Any optional information included (including conditions) must not compromise the validity of the required information.

Reference: **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.4

draft-sstc-bindings-model-09, section 4.2

Implementation notes: Test program implementing this test case establishes successful execution of the test case by inspection of the format of the returned assertion artefact.

4.2.9 Test Case 2-9: SOAP Profile: Valid Authorization Decision Assertion Returned in Valid Response to Valid Query.

Description: This test case uses the SOAP profile to request and receive an authorization decision assertion created by an implementation-under-test. It then confirms that the authorization decision assertion is valid for all required functionality.

Pass/Fail Criteria: Authorization decision assertion contains all required elements in the right format and sequence.

Requirements Reference: **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.4

draft-sstc-bindings-model-09, section 4.2

Implementation notes: Test program implementing this test case uses the SOAP profile to obtain the Authorization decision Assertion. It establishes successful execution of the test case by inspection of the format of the returned assertion.

4.3 Test Group 3 – Web Browser Profiles

The test cases in this test group check for conformance to the HTTP Web Browser Profiles for the SAML standard. Both the Browser/Artefact and Browser/POST profiles are optional. Any implementation or application claiming conformance to the Web Browser/Artefact Profile of SAML must be able to execute Test Cases 3-1 and 3-2 successfully. Any implementation or application claiming conformance to the Web Browser/Post Profile of SAML must be able to execute Test Cases 3-3 successfully.

4.3.1 Test Case 3-1: HTTP Web Browser/Artefact Profile: Valid Authentication Assertion Artefact Produced in Response to Valid Authentication Query.

Description: This test case submits an HTTP message to an implementation-under-test containing authentication credentials and checks that the implementation-under-test returns a valid authentication assertion artefact. It submits the authentication artefact to the implementation-under-test and confirms that the authentication assertion artefact has been properly consumed by inspecting the authentication assertion returned.

Pass/Fail Criteria: Authentication assertion artefact returned by implementation-under-test must be contain all required information in the right sequence and format. Any optional information included (including conditions) must not compromise the validity of the required information.

Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.3;

draft-sstc-bindings-model-09, section 4.1.1

Implementation notes: Test program implementing this test case establishes successful execution of the test case by inspection of the format of the returned assertion artefact.

4.3.2 Test Case 3-2: HTTP Web Browser/Artefact Profile: Valid Authentication Assertion Produced in Response to Valid Authentication Query with Artefact.

Description: This test case uses an artefact to request and receive an authenticatino assertion created by an implementation-under-test. It then confirms that the authentication assertion is valid for all required functionality.

Pass/Fail Criteria: Authorization decision assertion contains all required elements in the right format and sequence, AuthorizzationQuery is accepted by implementation-under-test, and AuthorizationResponse contains all required elements in correct sequence.

Requirements Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.3

draft-sstc-bindings-model-09, section 4.1.1

Implementation notes: Test program implementing this test case establishes successful execution of the test case by inspection of the format of the returned assertion.

4.3.3 Test Case 3-3: Web Browser/Post Profile: Valid Authentication Assertion Produced in Response to Valid Authentication Query.

Description: This test case submits an HTTP POST message to an implementation-under-test containing authentication credentials and checks that the implementation-under-test returns a valid authentication assertion.

Pass/Fail Criteria: Authentication assertion returned by implementation-under-test must contain all required information in the right sequence and format. Any optional information included (including conditions) must not compromise the validity of the required information.

Reference: **R-AUTHN**, and **R-MULTIDOMAIN**

Specification Reference: draft-sst-core-24, Section 2.4.3;

draft-sstc-bindings-model-09, section 4.1.2

Implementation notes: Test program implementing this test case establishes successful execution of the test case by inspection of the format of the returned assertion.

5 Test Suite

A test suite, which is the combination of test cases and test documentation, is used to check whether an implementation or application satisfies the requirements in the standard. The test cases, implemented by a test tool or a set of files (i.e., data, programs, scripts, or instructions for manual action) checks each requirement in the specification to determine whether the results produced by the implementation or application match the expected results, as defined by the specification.

The test documentation describes how the testing is to be done and the directions for the tester to follow. Additionally, the documentation should be detailed enough so that testing of a given implementation can be repeated with no change in test results.

Conformance testing is black box testing to test the functionality of an implementation. This means that the internal structure or the source code of a candidate implementation is not available to the tester. However, content and format of received or returned messages can be inspected as part of the determination of conformance.

The test suite for SAML should be platform independent, non-biased, objective tests. Generally a conformance test suite is a collection of combinations of legal and illegal inputs to the implementation being tested, together with a corresponding collection of expected results. Only the requirements specified in the standard are testable. A test suite should not check any implementation properties that are not described by the standard or set of standards. A test suite cannot require features that are optional in a standard, but if such features are present, a test suite could include tests for those features. A test suite does not assess the performance of an implementation unless performance requirements are specified in the specification, although implementation dependencies or machine dependencies may be demonstrated through the execution of the test cases.

The results of conformance testing apply only to the implementation and environment for which the tests are run. Test suites may be provided as a web-based system executed on a remote server, downloadable files for local execution, or a combination of remote and local access and execution. The method for providing and delivering the test suite depends on what is being tested as well as the objective for test suite use – that is, providing self-test capability or formal certification testing.

As a test suite for SAML becomes available, the following information will be provided:

- Reference Architecture
- Infrastructure
- Using the test suite
- Test result tabulation and reporting

The SAML test suite will be maintained on a best-effort basis.

6 Conformance Services

The OASIS Security Services Technical Committee does not itself provide conformance services. As the SAML test suite becomes avaialble and experience with SAML identified appropriate conformance testing approaches, the Conforrmance Specification will describe the services which the organization should provide including software services, releases, self-test kit, actual computer systems, facilities, web based interfaces, and availability.

7 References

[NIST/ITL] "What is this thing called conformance" [Rosenthal, Brady; NIST/ITL Bulletin,January 2001] http://www.itl.nist.gov/div897/ctg/conformance/bulletin-conformance.htm.

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[SAMLBind] P. Mishra et al., *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)*, http://www.oasis-open.org/committees/security/docs/draft-sstc-bindings-model-09.pdf, OASIS, December 2001.

[SAMLCore] P. Hallam-Baker et al., *Assertions and Protocol for the OASIS Security Assertion Markup Language*, http://www.oasis-open.org/committees/security/docs/draft-sstc-sec-core-24.pdf, OASIS, January 2002.

[SAMLGloss] J. Hodges et al., *Glossary for the OASIS Security Assertion Markup Language (SAML)*, http://www.oasis-open.org/committees/security/docs/draft-sstc-glossary-02.pdf, OASIS, December 2001.

[SAMLReqs] D. Platt et al., SAML Requirements and Use Cases, OASIS, December 2001.

[SAMLSec] C. McLaren et al., *Security Considerations for the OASIS Security Assertion Markup Language*, http://www.oasis-open.org/committees/security/docs/draft-sstc-sec-consider-03.pdf, OASIS, January 2002.

Appendix A. Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

Appendix B. Issues

Issue: Should any of the bindings or profiles be mandatory for all implementations or applications claiming conformance to the SAML standard?

Because of the importance of interoperability among implementations or applications claiming conformance to the SAML standard, one of the recommendations in this version of the SAML Conformance Specification is to require all implementations or applications to implement the SOAP binding for any assertions it supports (including in other profiles).. This ensures that 1) assertions created by the implementation or application can be retrieved using the SOAP binding, either directly or by means of an artefact, and can be inspected for validity; and 2) the ability of the implementation or application to consume assertions generated by another SAML-compliant implementation or application can be verified.

Alternatively, no single binding or profile need be mandatory, as long as an implementation or application claiming conformance is specific regarding which bindings and/or profiles it supports, with what assertions, and for what roles (producer / consumer). This is the approach taken in the Conformance Specification prior to verion 006.

Issue: Should the SOAP binding be mandatory?

The SOAP binding is suggested as mandatory because it provides the most fully-specified mechanism for requesting and returning all three assertions.

Issue: If the SOAP binding is mandatory, is it allowable to implement a subset of the assertions for that binding?

The current specification suggests that a subset of the SOAP binding (only the authentication assertion, for example) is allowable as satisfying this mandatory binding.