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

Document identifier: draft-sstc-conform-spec-12

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

Publication date: 22 March 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
Darren Platt, RSA
Irving Reid, Baltimore
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
08	Revised using bindings approach; corrected references; included issue
09	Removed SOAP Profile tests
10	Incorporated restriction for unbounded elements
11	Revised bounds for nested elements; mandatory/optional
12	Corrected test cases to correspond to Table 1

CONFORMANCE PROGRAM SPECIFICATION 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

2.5 MAXIMUM VALUES OF UNBOUNDED ELEMENTS 7

3 CONFORMANCE PROCESS 9

3.1 IMPLEMENTATION AND APPLICATION CONFORMANCE 9

3.2 PROCESS FOR DECLARING CONFORMANCE 10

4 TECHNICAL REQUIREMENTS FOR SAML CONFORMANCE 11

4.1 TEST GROUP 1 – SOAP OVER HTTP PROTOCOL BINDING 11

4.1.1 Test Case 1-1: SOAP Protocol Binding: Implementation-Under-Test Produces Valid Authentication Assertion in Valid Response to Authentication Query 11

4.1.2 Test Case 1-2: SOAP Protocol Binding: Implementation-Under-Test Consumes Valid Authentication Assertion, Requested in Valid Query 11

4.1.3 Test Case 1-3: SOAP Protocol Binding: Implementation-Under-Test Produces Valid Attribute Assertion in Valid Response to Attribute Query. 12

4.1.4 Test Case 1-4: SOAP Protocol Binding: Implementation-Under-Test Consumes Valid Attribute Assertion, Requested in Valid Query 12

4.1.5 Test Case 1-5: SOAP Protocol Binding: implemenation-Under-Test Produces Valid Authorization Decision Assertion in Valid Response to Authorization Decision Query. 12

4.1.6 Test Case 1-6: SOAP Protocol Binding: Implementation-Under-Test Consumes Valid Authorization Decision Assertion, Requested in Valid Query 13

4.2 TEST GROUP 2 – WEB BROWSER PROFILES 13

4.2.1 Test Case 2-1: HTTP Web Browser/Artifact Profile: Valid Authentication Assertion Produced in Response to Valid Authentication Query with Artifact. *13*

4.2.2 Test Case 2-2: HTTP Web Browser/Artifact Profile: Valid Authentication Assertion Request Corresponding to Valid Artifact Sent in valid HTTP message. *14*

4.2.3 Test Case 2-3: Web Browser/Post Profile: Valid Single Sign-on Assertion Received in Valid HTTP POST. *14*

4.2.4 Test Case 2-4: Web Browser/Post Profile: Valid Single Sign-on Assertion Sent in Valid HTTP POST. *14*

5 TEST SUITE **15**

6 CONFORMANCE SERVICES **16**

7 REFERENCES **17**

APPENDIX A. NOTICES **18**

APPENDIX B. ISSUES **19**

ISSUE: SHOULD ANY OF THE BINDINGS OR PROFILES BE MANDATORY FOR ALL IMPLEMENTATIONS OR APPLICATIONS CLAIMING CONFORMANCE TO THE SAML STANDARD? 19

ISSUE: SHOULD THE SOAP BINDING BE MANDATORY? 19

ISSUE: IF THE SOAP BINDING IS MANDATORY, IS IT ALLOWABLE TO IMPLEMENT A SUBSET OF THE ASSERTIONS FOR THAT BINDING? 19

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 white papers, 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 specifications.

SAML conformance must be expressed in terms of which SAML bindings and profiles are supported by a given application or implementation. The application or implementation claiming conformance to the SAML standard must support the SOAP protocol binding for at least one assertion. An application or implementation may also support the web browser profiles.

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 requester or responder or both requester 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 binding or profile to which an application or implementation claims conformance, the claim must stipulate whether the requester and/or responder roles are supported and for which assertions for those roles.

For example, an implementation consisting solely of an Authentication Authority responsible for generating Authentication Assertions and returning those assertions in response to a SOAP-over-HTTP request for assertion would correspond to the cell in the third column of the second row (including the column title row). If the implementation also supported the return of the assertion in the Browser/Artifact profile, then the third column in the fifth row would also be supported.

149 **Table 1: Protocol Bindings and Profiles for SAML Assertions**

Binding or Profile	Consumer Role	Producer Role
SOAP over HTTP protocol binding	Send an Authentication Query to request an Authentication Assertion from a producer; consume the returned assertion.	Produce an Authentication Assertion; and return an AuthenticationResponse containing the assertion to the consumer.
	Send an AttributeQuery to request an Attribute Assertion from a roducer; consume the returned assertion.	Produce an Attribute Assertion; and return an AttributeResponse containing the assertion to the consumer.
	Send an AuthorizationDecisionQuery to request an Authorization Decision Assertion from a roducer; consume the returned assertion.	Produce an Authorization Decision Assertion; and return AuthorizationDecisionResponse containing the assertion to the consumer.
Browser/Artifact Profile	Receive an artifact corresponding to an Authentication Assertion; request the corresponding assertion; and consume the returned assertion.	Produce and send an artifact to a consumer; produce the corresponding Authentication Assertion; and on request containing the artifact, return the assertion to the consumer.
Browser/POST Profile	Receive a Single-Signon Assertion in a POST message and consume the assertion	Produce the Single-Signon Assertion

150

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

153 *[Application or implementation] as both requester and responder supports all SAML protocol bindings and*
154 *profiles, for all assertions and required elements. No optional elements for the assertions, bindings and*
155 *profiles are implemented.*

156 *[Application or implementation] as both requester and responder supports the SOAP protocol binding for all*
157 *assertions. It also supports the Conditions optional elements for all assertions in the SOAP protocol*
158 *binding. It does not support the Web Browser profiles for any assertion.*

159 *[Application or implementation] as both requester and responder supports the SOAP protocol binding for all*
160 *assertions, for all assertions.. It also support the Web Browser Profile for Authentication Assertion and all*
161 *required elements. No optional elements for the assertions, bindings and profiles are implemented.*

162 An application or implementation that claims conformance for a particular binding or profile must support all
163 required elements of that binding or profile and of the assertions supported with that binding or profile. It
164 must also state which assertions are supported and which, if any optional elements for that binding or
165 profile and corresponding assertions are supported.

166 2.3 Mandatory/Optional Elements in SAML Conformance

167 The SOAP protocol binding must be implemented by all implementations or applications claiming SAML
168 conformance, for each assertion claimed as supported through a binding or profile. (see Appendix B:
169 Issues)

The SAML schema and binding specifications include both mandatory and optional elements. A conforming application or implementation must be able to handle all valid SAML elements, including those that are optional. However, it does not have to produce those optional elements.

For example:

- An application or implementation that consumes assertions must be able to handle assertions that include the optional “condition” element, such as by rejecting any conditions that it does not recognize.
- An application or implementation that produces assertions can, but is not required to, include the optional “condition” element in those assertions.
- An application or implementation claiming support for an assertion must support the SOAP over HTTP protocol binding. It can also, optionally, implement the protocol by means of another binding.

The test cases for SAML conformance are intended to check for support of all valid SAML elements. 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 profiles 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.

2.5 Maximum Values of Unbounded Elements

The SAML schema supports a number of elements that can be specified multiple times in an assertion, request or response. An application or implementation claiming conformance must support at least the values listed in Table 2 below for each of the elements defined as “unbounded” in the SAML schema. In those cases where the maximum value is greater than the listed values, the application or implementation should state what that maximum supported value is.

However. Some of the elements in the table can be nested, such that repeated elements have a multiplicative effect on the number of elements. For example, trees of nested unbounded elements include the following:

214 Response > Assertion > Signature

215 Response > Assertion > Advice

216 Response > Assertion > Condition > Target

217 Response > Assertion > Condition > Audience

218 Response > Assertion > Statement > SubjectConfirmationMethod

219 Response > Assertion > Statement > AuthorityBinding

220 Response > Assertion > Statement > Action

221 Response > Assertion > Statement > Attribute > AttributeValue

222 In a response containing 10 assertions, each with 10 AttributeStatements, each with 10 Attributes, each
223 with 10 AttributeValues, this tree alone comprises 10,000 elements.

224 Therefore, In order to minimize the potential impact of nested unbounded elements, an application or
225 implementation can limit the total number of elements supported in a given request, response or (when
226 this is used in the POST profile) assertion to no more than 1000 total elements and still claim conformance
227 to the SAML V1.0 specification.

228 **Table 2: Unbounded Elements**

Element	Parent Element	Maximum Value	Section in sstc-core
Statement	Assertion	1000	2.3.3
Signature	Assertion	1000	2.3.3
Condition	Assertion	1000	2.3.3
Audience	Condition	1000	2.3.3.1.3
Target	Condition	1000	2.3.3.1.4
Advice	Assertion	1000	2.3.3.2
ConfirmationMethod	SubjectConfrmation	1000	2.4.2.3
AuthorityBinding	AuthenticationStatement	1000	2.4.3.2
Evidence	AuthorizationDecisionStatement	1000	2.4.4
Actions	Action	1000	2.4.4.1
Attribute	AttributeStatement	1000	2.4.5
AttributeValue	Attribute	1000	2.4.5.1.1
RespondWith	Request	1000	3.2.1
AssertionArtifact	Request	1000	3.2.2
AttributeDesignator	AttributeQuery	1000	3.3.4
Evidence	AuthorizationDecisionQuery	1000	3.3.5
Assertion	Response	1000	3.4.2
StatusMessage	Status	1000	3.4.3
StatusDetail	Status	1000	3.4.3

229

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 asserting that conformance testing has been performed.

The conformance process does not stipulate whether validation is performed by the implementer, 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 technical criteria which apply to declaring conformance to the SAML standard. The requirements are specified as test cases, corresponding to the 10 possible subsets of conformance defined in Table 1 above.

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 for the claimed assertion or assertions and role (producer or consumer), even if support for this protocol binding is incidental to the primary purposes of the application or implementation.

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

Description: This test case requests and receives an authentication assertion created by an implementation-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: SAML Core, sections 2.4.3 and 3

SAML Bind, section 3.1.

Implementation notes: The implementation-under-test executes the authentication assertion producer role.

4.1.2 Test Case 1-2: SOAP Protocol Binding: Implementation-Under-Test Consumes Valid Authentication Assertion, Requested in Valid Query

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

Pass/Fail Criteria: AuthenticationQuery contains all required elements in the right format and sequence; authentication response and assertion are consumed.

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

Specification Reference: SAML Core, sections 2.4.3 and 3

SAML Bind, section 3.1

Implementation notes: The implementation-under-test executes the authentication assertion consumer role. Test program and implementation-under-test must agree how to validate that assertion was consumed.

4.1.3 Test Case 1-3: SOAP Protocol Binding: Implementation-Under-Test Produces Valid Attribute Assertion in Valid Response to 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 returned by the implementation-under-test 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: SAML Core, Sections 2.4.5 and 3

SAML Bind, section 3.1.

Implementation notes: The implementation-under-test executes the attribute assertion producer role.

4.1.4 Test Case 1-4: SOAP Protocol Binding: Implementation-Under-Test Consumes Valid Attribute Assertion, Requested in Valid Query

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

Pass/Fail Criteria: AttributeQuery contains all required elements in the right format and sequence; attribute response and assertion are consumed.

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

Specification Reference: SAML Core, sections 2.4.5 and 3

SAML Bind, section 3.1

Implementation notes: The implementation-under-test executes the attribute assertion consumer role. Test program and implementation-under-test must agree how to validate that assertion was consumed.

4.1.5 Test Case 1-5: SOAP Protocol Binding: implemenation-Under-Test Produces Valid Authorization Decision Assertion in Valid Response to Authorization Decision Query.

Description: This test case requests and receives an authentication assertion created by an implementation-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: SAML Core, Section 2.4.4 and 3

SAML Bind, section 3.1.

Implementation notes: The implementation-under-test executes the authorization decision assertion producer role.

4.1.6 Test Case 1-6: SOAP Protocol Binding: Implementation-Under-Test Consumes Valid Authorization Decision Assertion, Requested in Valid Query

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

Pass/Fail Criteria: AuthorizationQuery contains all required elements in the right format and sequence; authorization decision response and assertion are consumed.

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

Specification Reference: SAML Core, sections 2.4.4 and 3

SAML Bind, section 3.1

Implementation notes: The implementation-under-test executes the authorization decision assertion consumer role. Test program and implementation-under-test must agree how to validate that assertion was consumed.

4.2 Test Group 2 – 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/Artifact and Browser/POST profiles are optional. Any implementation or application claiming conformance to the Web Browser/Artifact Profile of SAML must be able to execute Test Case 2-1 successfully for the assertion producer role and/or Test Case 2-2 successfully for the assertion consumer role. Any implementation or application claiming conformance to the Web Browser/Post Profile of SAML must be able to execute Test Case 2-3 successfully for the assertion producer role and/or Test Case 2-4 successfully for the assertion consumer role.

4.2.1 Test Case 2-1: HTTP Web Browser/Artifact Profile: Valid Authentication Assertion Produced in Response to Valid Authentication Query with Artifact.

Description: This test case receives an artifact in a valid HTTP message from an implementation-under-test. The test case confirms the artifact is valid for all required functionality. It then uses the artifact in the SOAP protocol binding to request and receive an authentication assertion created by an implementation-under-test corresponding to the artifact. 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, AuthorizationQuery is accepted by implementation-under-test, and AuthorizationResponse contains all required elements in correct sequence.

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

Specification Reference: SAML Core, Section 2.4.3

SAML Bind, section 4.1.1

Implementation notes: Test program performs the destination site (consumer) operations for the profile; implementation-under-test performs source site (producer) operations.

4.2.2 Test Case 2-2: HTTP Web Browser/Artifact Profile: Valid Authentication Assertion Request Corresponding to Valid Artifact Sent in valid HTTP message.

Description: This test case sends a valid artifact in a valid HTTP message to an implementation-under-test. The test case then receives an authentication query containing the artifact from the implementation-under-test. It confirms that the authentication query is valid for all required functionality, then returns the authentication assertion to the implementation-under-test, and confirms that the assertion was consumed.

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

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

Specification Reference: SAML Core, Section 2.4.3

SAML Bind, section 4.1.1

Implementation notes: Test program performs the source site (producer) operations for the profile; implementation-under-test performs destination site (consumer) operations.

4.2.3 Test Case 2-3: Web Browser/Post Profile: Valid Single Sign-on Assertion Received in Valid HTTP POST.

Description: This test case receives an HTTP POST message from an implementation-under-test containing a Single Sign-on assertion and checks that the assertion is valid.

Pass/Fail Criteria: Authentication assertion sent 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: SAML Core, Section 2.4.3;

SAML Bind, section 4.1.2

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

4.2.4 Test Case 2-4: Web Browser/Post Profile: Valid Single Sign-on Assertion Sent in Valid HTTP POST.

Description: This test case sends an HTTP POST message to an implementation-under-test containing a Single Sign-on assertion and checks that the assertion is consumed.

Pass/Fail Criteria: Implementation-under-test allows access based on authentication assertion it receives and consumes.

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

Specification Reference: SAML Core, Section 2.4.3;

SAML Bind, section 4.1.2

Implementation notes: Test program (producer role) and implementation-under-test must agree how to validate that access is allowed.

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 available and experience with SAML identified appropriate conformance testing approaches, the Conformance 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-12.pdf, OASIS, March 2002.

[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-28.pdf, OASIS, March 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-04.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' 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 implementers 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 artifact, 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 (responder / requester). This is the approach taken in the Conformance Specification prior to version 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.