



Conformance Program Specification for the OASIS Security Assertion Markup Language (SAML) V1.1

Last Call Working Draft 03, 2 May 2003

Document identifier:

sstc-saml-conform-1.1-draft-03

Location:

http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

Editors:

Robert Griffin, Entrust (robert.griffin@entrust.com)
Eve Maler, Sun Microsystems (eve.maler@sun.com)
Robert Philpott, RSA Security (rphilpott@rsasecurity.com)

Contributors:

Irving Reid, Baltimore Technologies
Krishna Sankar, Cisco Systems
Hal Lockhart, BEA Systems (formerly of Entegriy Solutions)
Marc Chanliau, Netegrity
Prateek Mishra, Netegrity
Lynne Rosenthal, NIST
Mark Skall, NIST
Darren Platt, formerly with RSA Security
Charles Norwood, SAIC
Emily Xu, Sun Microsystems
Sai Allarvarpu, Sun Microsystems
Mike Myers, Traceroute Security
Mark O'Neill, Vordel
Tony Palmer, Vordel

Abstract:

This specification describes the program and technical requirements for SAML conformance.

Status:

This document is a **last-call working draft** of the OASIS Security Services Technical Committee. We solicit your comments; they must be received by Friday, 16 May 2003 in order for the committee to consider them for inclusion in the Committee Specification.

If you are on the security-services@lists.oasis-open.org list for committee members, send comments there. If you are not on that list, subscribe to the security-services-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email

38 message to security-services-comment-request@lists.oasis-open.org with the word "subscribe"
39 as the body of the message.
40 For information on whether any patents have been disclosed that may be essential to
41 implementing this specification, and any offers of patent licensing terms, please refer to the
42 Intellectual Property Rights section of the Security Services TC web page ([http://www.oasis-](http://www.oasis-open.org/committees/security/)
43 [open.org/committees/security/](http://www.oasis-open.org/committees/security/)).

Table of Contents

45	1	Introduction.....	5
46	1.1	Scope of the Conformance Program	5
47	1.2	Notation.....	5
48	2	Conformance Clause.....	6
49	2.1	SAML Specification Set	6
50	2.2	Declaration of SAML Conformance	6
51	2.3	Mandatory/Optional Elements in SAML Conformance	8
52	2.4	Impact of Extensions on SAML Conformance	8
53	2.5	Maximum Values of Unbounded Elements.....	9
54	3	Conformance Process.....	11
55	3.1	Implementation and Application Conformance	11
56	3.2	Process for Declaring Conformance.....	12
57	4	Technical Requirements for SAML Conformance.....	13
58	4.1	Test Group 1 – SOAP over HTTP Protocol Binding	13
59	4.1.1	Test Case 1-1: SOAP Binding: Implementation-Under-Test Produces Valid Authentication Assertion in Valid Response to Authentication Query	13
60	4.1.2	Test Case 1-2: SOAP Binding: Implementation-Under-Test Consumes Valid Authentication Assertion, Requested in Valid Query.....	14
61	4.1.3	Test Case 1-3: SOAP Binding: Implementation-Under-Test Produces Valid Attribute Assertion in Valid Response to Attribute Query.....	14
62	4.1.4	Test Case 1-4: SOAP Binding: Implementation-Under-Test Consumes Valid Attribute Assertion, Requested in Valid Query.....	14
63	4.1.5	Test Case 1-5: SOAP Binding: Implementation-Under-Test Produces Valid Authorization Decision Assertion in Valid Response to Authorization Decision Query	15
64	4.1.6	Test Case 1-6: SOAP Binding: Implementation-Under-Test Consumes Valid Authorization Decision Assertion, Requested in Valid Query.....	15
65	4.2	Test Group 2 – Web Browser SSO Profiles	15
66	4.2.1	Test Case 2-1: Browser/Artifact Profile: Valid Authentication Assertion Produced in Response to Valid Authentication Query with Artifact.....	15
67	4.2.2	Test Case 2-2: Browser/Artifact Profile: Valid Authentication Assertion Request Corresponding to Valid Artifact Sent in Valid HTTP Message	16
68	4.2.3	Test Case 2-3: Browser/POST Profile: Valid SSO Assertion Received in Valid HTTP POST ..	16
69	4.2.4	Test Case 2-4: Browser/Post Profile: Valid SSO Assertion Sent in Valid HTTP POST	16
70	5	Test Suite	18
71	6	Conformance Services	19
72	7	References	20
73		Appendix A. Acknowledgments.....	21

82 Appendix B. Notices 22
83 Appendix C. Revision History 23
84

85 1 Introduction

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

87 1.1 Scope of the Conformance Program

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

92 The deliverables of the SAML conformance effort include:

- 93 • Conformance clause, defining at a high level what conformance means for the SAML standard.
- 94 • Conformance program specification, defining how an implementation or application establishes
95 conformance.
- 96 • Input to the creation of a conformance test suite. This is a high-level specification for a set of test
97 programs, result files, and report generation tools that can be used by vendors of SAML-compliant
98 software, buyers interested in confirming SAML compliance of software, and testing labs running
99 conformance tests on behalf of vendors or buyers.

100 Section 2 of this document provides the SAML Conformance Clause. Section 3 deals with defining and
101 specifying the process by which conformance to the SAML specification set can be demonstrated and
102 certified. Section 4 elucidates the technical requirements that constitute conformance; this includes both
103 the levels of conformance that can be demonstrated and the requirements for each of those levels of
104 conformance. Section 5 describes what a test suite for SAML should include. Section 6 defines the
105 services that may become available to assist in establishing conformance. Section 7 gives information for
106 documents referenced in this specification.

107 1.2 Notation

108 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
109 NOT", "RECOMMENDED", "DOES", and "OPTIONAL" in this specification are to be interpreted as
110 described in IETF RFC 2119 [**RFC2119**]:

111 ...they **MUST** only be used where it is actually required for interoperation or to limit behavior
112 which has potential for causing harm (e.g., limiting retransmissions)...

113 These keywords are thus capitalized when used to unambiguously specify requirements over protocol and
114 application features and behavior that affect the interoperability and security of implementations. When
115 these words are not capitalized, they are meant in their natural-language sense.

116 2 Conformance Clause

117 The objectives of the SAML Conformance Clause are to:

- 118 • Ensure a common understanding of conformance and what is required to claim conformance
- 119 • Promote interoperability in the exchange of authentication and authorization information
- 120 • Promote uniformity in the development of conformance tests

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

123 2.1 SAML Specification Set

124 The following four specifications, in addition to this SAML conformance program specification, comprise
125 the Version 1.1 specification set for SAML:

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

130 The SAML Core document also references the schema definitions for SAML assertions and protocols:

- 131 • Assertion schema [**SAMLAssertion**]
- 132 • Protocol schema [**SAMLProtocol**]

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

136 2.2 Declaration of SAML Conformance

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

141 SAML conformance **MUST** be expressed in terms of which SAML bindings and profiles are supported by
142 a given application or implementation. The application or implementation claiming conformance to the
143 SAML standard **MUST** support the SOAP protocol binding for assertions containing at least one statement
144 type. An application or implementation **MAY** also support the web browser profiles.

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

- 147 • Whether the application or implementation acts as producer, consumer, or both producer and
148 consumer of the SAML messages in the supported bindings and profiles.
- 149 • Which assertions and statements the application or implementation supports for each supported
150 binding.

151 Table 1 shows the protocols, protocol bindings, and profiles applicable to each SAML assertion/statement
 152 type. For each SAML binding or profile to which an application or implementation claims conformance, the
 153 claim MUST stipulate whether the producer and/or consumer roles are supported and for which assertions
 154 and statements for those roles. (Note that the OASIS Web Services Security Technical Committee has
 155 produced a draft “SAML token profile” of the WSS specification [WSS-SAML], which describes how to
 156 use SAML assertions to secure a web service message. This specification does not discuss conformance
 157 to that profile of SAML.)

158 For example, an implementation consisting solely of an authentication authority responsible for generating
 159 assertions containing authentication statements and returning those assertions in response to a SOAP-
 160 over-HTTP request for assertion would correspond to the “producer role” for the SOAP over HTTP
 161 binding. If the implementation also supported the return of the assertion in the browser/artifact profile, then
 162 the “producer role” for that profile would also be supported.

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

Binding or Profile	Consumer Role	Producer Role
SOAP over HTTP protocol binding	Send a request containing an authentication query to request an assertion containing an authentication statement from a producer; consume the returned assertion.	Produce an assertion containing an authentication statement and return a response containing the assertion to the consumer.
	Send a request containing an attribute query to request an assertion containing an attribute statement from a producer; consume the returned assertion.	Produce an assertion containing an attribute statement and return a response containing the assertion to the consumer.
	Send an AuthorizationDecisionQuery to request an assertion containing an authorization decision statement from a producer; consume the returned assertion.	Produce an assertion containing an authorization decision statement and return a response containing the assertion to the consumer.
Browser/Artifact Profile	Receive an artifact corresponding to a single sign-on (SSO) assertion; request the corresponding assertion; and consume the returned assertion.	Produce and send an artifact to a consumer; produce the corresponding SSO assertion; and on request containing the artifact, return the assertion to the consumer.
Browser/POST Profile	Receive a SSO assertion in a POST message and consume the assertion.	Produce the SSO assertion.

164
 165 An application or implementation should express its level of conformance in terminology such as the
 166 following:

167 [Application or implementation] as both producer and consumer supports all SAML protocol
 168 bindings and profiles, for all assertions, statements, and required elements. No optional
 169 elements for the assertions, statements, bindings, and profiles are produced.

170 [Application or implementation] as both producer and consumer supports the SOAP protocol
 171 binding for all queries, assertions, and statements. It produces the <Conditions> optional

172 elements for all assertions in the SOAP protocol binding. It does not support the browser
173 profiles for any assertion.

174 [Application or implementation] as both producer and consumer supports the SOAP protocol
175 binding for all assertions and statements. It also supports the browser/artifact profile and all
176 required elements. No optional elements for the assertions, statements, bindings, and profiles
177 are produced.

178 An application or implementation that claims conformance for a particular binding or profile MUST support
179 all required elements of that binding or profile and of the assertions supported with that binding or profile.
180 It MUST also state which assertions and statements are supported and which, if any, optional elements for
181 that binding or profile and corresponding assertions and statements are supported.

182 **2.3 Mandatory/Optional Elements in SAML Conformance**

183 The SOAP protocol binding MUST be implemented by all implementations or applications claiming SAML
184 conformance, for each assertion and statement type claimed as supported through a binding or profile.

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

188 For example:

- 189 • An application or implementation that consumes assertions must be able to handle assertions that
190 include the optional <Condition> element, such as by rejecting any conditions that it does not
191 recognize.
- 192 • An application or implementation that produces assertions may, but is not required to, include the
193 optional <Condition> element in those assertions.
- 194 • An application or implementation claiming support for an assertion must support the SOAP over HTTP
195 protocol binding. It can also, optionally, implement the protocol by means of another binding.

196 The test cases for SAML conformance are intended to check for support of all valid SAML elements. They
197 also check whether an implementation or application accepts and properly handles optional assertion
198 elements (such as <Condition>) whose value the implementation or application does not recognize.

199 **2.4 Impact of Extensions on SAML Conformance**

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

- 203 • Extensions MUST NOT re-define semantics for existing functions.
- 204 • Extensions MUST NOT alter the specified behavior of interfaces defined in the SAML specification
205 set.
- 206 • Extensions MAY add additional behaviors.
- 207 • Extensions MUST NOT cause standard-conforming functions (i.e., functions that do not use the
208 extensions) to execute incorrectly.

209 SAML bindings and profiles MAY be extended so long as the above conditions are met. If a system is
210 extending SAML assertions or statements:

- 211 • The mechanism for determining application conformance and the extensions MUST be clearly
212 described in the documentation, and the extensions MUST be marked as such;
 - 213 • Extensions MUST follow the spirit, principles, and guidelines of the SAML specification set, that is, the
214 specifications MUST be extended in a standard manner as defined in the extension fields.
 - 215 • In the case where an implementation has added additional behaviors, the implementation MUST
216 provide a mechanism whereby a conforming application shall be recognized as such, and be
217 executed in an environment that supports the functional behavior defined in this specification set.
- 218 Extensions are outside the scope of conformance. There are no mechanisms specified to validate and
219 verify the extensions.

220 2.5 Maximum Values of Unbounded Elements

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

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

- 229 Response > Assertion > Signature
- 230 Response > Assertion > Advice
- 231 Response > Assertion > Condition > Target
- 232 Response > Assertion > Condition > Audience
- 233 Response > Assertion > Statement > SubjectConfirmationMethod
- 234 Response > Assertion > Statement > AuthorityBinding
- 235 Response > Assertion > Statement > Action
- 236 Response > Assertion > Statement > Attribute > AttributeValue

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

239 Therefore, in order to minimize the potential impact of nested unbounded elements, an application or
240 implementation MAY limit the total number of elements supported in a given request, response or (when
241 this is used in the POST profile) assertion to no more than 1000 total elements and still claim
242 conformance to the SAML V1.1 specification set.

243 **Table 2: Unbounded Elements**

Element	Parent Element	Maximum Value
Statement	Assertion	1000
Signature	Assertion	1000
Condition	Assertion	1000
Audience	Condition	1000
Target	Condition	1000
Advice	Assertion	1000
ConfirmationMethod	SubjectConfirmation	1000
AuthorityBinding	AuthenticationStatement	1000

Element	Parent Element	Maximum Value
Evidence	AuthorizationDecisionStatement	1000
Actions	Action	1000
Attribute	AttributeStatement	1000
AttributeValue	Attribute	1000
RespondWith	Request	1000
AssertionArtifact	Request	1000
AttributeDesignator	AttributeQuery	1000
Evidence	AuthorizationDecisionQuery	1000
Assertion	Response	1000
StatusMessage	Status	1000
StatusDetail	Status	1000

244

245

3 Conformance Process

246
247

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

248
249
250
251
252
253

- **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. Conformance testing performed by implementors 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 a certification process.

254
255
256

- **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.

257
258
259
260
261

- **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.

262
263
264
265
266
267

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 implementor who has validated SAML conformance by means of conformance testing **MUST NOT** 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.

268
269
270
271
272
273
274

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.

275
276
277

The Security Services 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, included in this specification.

278
279
280
281

The test cases can be used to create a test suite that can be run against an implementation to demonstrate any of the several levels of conformance defined in the conformance clause of the SAML specification. The Security Services Technical Committee is not responsible for developing the test suite nor for testing of particular implementations.

282

3.1 Implementation and Application Conformance

283

SAML Conformance is applicable to:

284
285
286

- Implementations of SAML assertions, statements, 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.

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

289 A conforming **implementation** MUST meet all the following criteria:

- 290 1. The implementation MUST support all the required interfaces defined within the specification set for a
291 given binding or profile. It MUST also specify which assertions and statements relevant to that binding
292 or profile are supported. The implementation MUST support the functional behavior described in the
293 specification.
- 294 2. The implementation MAY provide additional or enhanced facilities not required by this specification
295 set. These nonstandard extensions MUST NOT alter the specified behavior of interfaces defined in
296 this specification. They MAY add additional behaviors. In these circumstances, the implementation
297 MUST provide a mechanism whereby a SAML conforming application shall be recognized as such,
298 and be executed in an environment that supports the functional behavior defined in this specification
299 set.

300 A conforming **application** MUST meet all the following criteria:

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

307 **3.2 Process for Declaring Conformance**

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

- 310 1. Determine which bindings and protocols will be asserted as conforming.
- 311 2. Implement the test suite for the conformance tests relevant to the conformance being claimed.
- 312 3. Validate the application or implementation by executing those conformance tests.
- 313 4. Send the statement claiming conformance to the Security Services Technical Committee so that it can
314 be posted on the SAML web site. A statement of any bindings and profiles being used that are not part
315 of the SAML standard should also be sent to the Security Services Technical Committee at the same
316 time for posting on the SAML web site.

317

4 Technical Requirements for SAML Conformance

318 This section defines the technical criteria that apply to declaring conformance to the SAML standard. The
319 requirements are specified as test cases, corresponding to the 10 possible subsets of conformance
320 defined in Table 1.

321 Each test case includes:

- 322 • A description of the test purpose (that is, what is being tested – the conditions, requirements, or
323 capabilities which are to be addressed by a particular test)
- 324 • The pass/fail criteria
- 325 • A reference to the requirement in the requirements document relevant to the test case
- 326 • A reference to the section in the specification set from which the test case is derived (that is,
327 traceability back to the specification)

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

330 4.1 Test Group 1 – SOAP over HTTP Protocol Binding

331 The test cases in this test group check for conformance to the SAML SOAP protocol binding. Any
332 implementation or application claiming conformance to SAML MUST be able to execute these test cases
333 successfully for the claimed assertion or assertions and role (producer or consumer), even if support for
334 this protocol binding is incidental to the primary purposes of the application or implementation.

335 For convenience, assertions containing an authentication statement will be referred to in this section as
336 *authentication assertions*, assertions containing an attribute statement as *attribute assertions*, and
337 assertions containing an authorization decision statement as *authorization decision assertions*.

338 4.1.1 Test Case 1-1: SOAP Binding: Implementation-Under-Test Produces 339 Valid Authentication Assertion in Valid Response to Authentication 340 Query

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

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

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

349 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 3.1

350 **Implementation Notes:** The implementation-under-test executes the authentication assertion producer
351 role.

352 **4.1.2 Test Case 1-2: SOAP Binding: Implementation-Under-Test Consumes**
353 **Valid Authentication Assertion, Requested in Valid Query**

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

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

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

361 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 3.1

362 **Implementation Notes:** The implementation-under-test executes the authentication assertion consumer
363 role. It is up to the test program and implementation-under-test to determine how to validate that assertion
364 was consumed.

365 **4.1.3 Test Case 1-3: SOAP Binding: Implementation-Under-Test Produces**
366 **Valid Attribute Assertion in Valid Response to Attribute Query**

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

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

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

374 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 3.1

375 **Implementation Notes:** The implementation-under-test executes the attribute assertion producer role.

376 **4.1.4 Test Case 1-4: SOAP Binding: Implementation-Under-Test Consumes**
377 **Valid Attribute Assertion, Requested in Valid Query**

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

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

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

385 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 3.1

386 **Implementation Notes:** The implementation-under-test executes the attribute assertion consumer role. It
387 is up to the test program and implementation-under-test to determine how to validate that assertion was
388 consumed.

389 **4.1.5 Test Case 1-5: SOAP Binding: Implementation-Under-Test Produces**
390 **Valid Authorization Decision Assertion in Valid Response to**
391 **Authorization Decision Query**

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

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

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

400 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 3.1

401 **Implementation Notes:** The implementation-under-test executes the authorization decision assertion
402 producer role.

403 **4.1.6 Test Case 1-6: SOAP Binding: Implementation-Under-Test Consumes**
404 **Valid Authorization Decision Assertion, Requested in Valid Query**

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

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

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

412 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 3.1

413 **Implementation Notes:** The implementation-under-test executes the authorization decision assertion
414 consumer role. It is up to the test program and implementation-under-test to determine how to validate
415 that assertion was consumed.

416 **4.2 Test Group 2 – Web Browser SSO Profiles**

417 The test cases in this test group check for conformance to the web browser single sign-on (SSO) profiles
418 of the SAML standard. Both the browser/artifact and browser/POST profiles are optional. Any
419 implementation or application claiming conformance to the browser/artifact profile MUST be able to
420 execute Test Case 2-1 successfully for the assertion producer role and/or Test Case 2-2 successfully for
421 the assertion consumer role. Any implementation or application claiming conformance to the
422 browser/POST profile MUST be able to execute Test Case 2-3 successfully for the assertion producer role
423 and/or Test Case 2-4 successfully for the assertion consumer role.

424 **4.2.1 Test Case 2-1: Browser/Artifact Profile: Valid Authentication Assertion**
425 **Produced in Response to Valid Authentication Query with Artifact**

426 **Description:** This test case receives an artifact in a valid HTTP message from an implementation-under-
427 test. The test case confirms the artifact is valid for all required functionality. It then uses the artifact in the
428 SOAP protocol binding to request and receive an authentication assertion created by an implementation-

429 under-test corresponding to the artifact. It then confirms that the authentication assertion is valid for all
430 required functionality.

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

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

435 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 4.1.1

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

438 **4.2.2 Test Case 2-2: Browser/Artifact Profile: Valid Authentication Assertion** 439 **Request Corresponding to Valid Artifact Sent in Valid HTTP Message**

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

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

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

446 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 4.1.1

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

449 **4.2.3 Test Case 2-3: Browser/POST Profile: Valid SSO Assertion Received in** 450 **Valid HTTP POST**

451 **Description:** This test case receives an HTTP POST message from an implementation-under-test
452 containing an SSO assertion and checks that the assertion is valid.

453 **Pass/Fail Criteria:** Authentication assertion sent by implementation-under-test MUST contain all required
454 information in the right sequence and format. Any optional information included (including conditions)
455 MUST NOT compromise the validity of the required information.

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

457 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 4.1.2

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

460 **4.2.4 Test Case 2-4: Browser/Post Profile: Valid SSO Assertion Sent in Valid** 461 **HTTP POST**

462 **Description:** This test case sends an HTTP POST message to an implementation-under-test containing
463 an SSO assertion and checks that the assertion is consumed.

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

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

467 **Specification Reference:** [SAMLCore] Sections 2.3, 2.4, and 3; [SAMLBind] Section 4.1.2

468 **Implementation Notes:** It is up to the test program and implementation-under-test to determine how to
469 validate that assertion was consumed.

470

5 Test Suite

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

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

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

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

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

497

6 Conformance Services

498 The OASIS Security Services Technical Committee does not itself provide conformance services. As
499 SAML test suites become available and experience with SAML identified appropriate conformance testing
500 approaches, the Conformance Specification will describe the services which a conformance services
501 organization should provide, including software services, releases, self-test kit, actual computer systems,
502 facilities, web based interfaces, and availability.

503

7 References

- 504 **[NIST/ITL]** “*What is this thing called conformance*” [Rosenthal, Brady; NIST/ITL Bulletin,
505 January 2001] [http://www.itl.nist.gov/div897/ctg/conformance/bulletin-](http://www.itl.nist.gov/div897/ctg/conformance/bulletin-conformance.htm)
506 [conformance.htm](http://www.itl.nist.gov/div897/ctg/conformance/bulletin-conformance.htm).
- 507 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
508 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 509 **[SAMLAssertion]** Phillip Hallam-Baker et al., *Assertions Schema for the OASIS Security Assertion*
510 *Markup Language (SAML)*, <http://www.oasis-open.org/committees/security/>,
511 OASIS, May 2003.
- 512 **[SAMLBind]** Prateek Mishra et al., *Bindings and Profiles for the OASIS Security Assertion*
513 *Markup Language (SAML)*, <http://www.oasis-open.org/committees/security/>,
514 OASIS, May 2003.
- 515 **[SAMLCore]** Phillip Hallam-Baker et al., *Assertions and Protocol for the OASIS Security*
516 *Assertion Markup Language (SAML)*, [http://www.oasis-](http://www.oasis-open.org/committees/security/)
517 [open.org/committees/security/](http://www.oasis-open.org/committees/security/), OASIS, May 2003.
- 518 **[SAMLGloss]** Jeff Hodges et al., *Glossary for the OASIS Security Assertion Markup Language*
519 *(SAML)*, <http://www.oasis-open.org/committees/security/>, OASIS, May 2003.
- 520 **[SAMLProtocol]** Phillip Hallam-Baker et al., *Protocol Schema for the OASIS Security Assertion*
521 *Markup Language (SAML)*, <http://www.oasis-open.org/committees/security/>,
522 OASIS, May 2003.
- 523 **[SAMLReqs]** Darren Platt et al., *SAML Requirements and Use Cases*, OASIS, April 2002.
- 524 **[SAMLSec]** Chris McLaren et al., *Security Considerations for the OASIS Security Assertion*
525 *Markup Language (SAML)*, <http://www.oasis-open.org/committees/security/>,
526 OASIS, May 2003.
- 527 **[WSS-SAML]** P. Hallam-Baker et al., *Web Services Security: SAML Token Profile*, OASIS,
528 March 2003, <http://www.oasis-open.org/committees/wss>.

529 Appendix A. Acknowledgments

530 The editors would like to acknowledge the contributions of the OASIS SAML Technical Committee, whose
531 voting members at the time of publication were:

- 532 • Irving Reid, Baltimore Technologies
- 533 • Hal Lockhart, BEA Systems
- 534 • Ronald Jacobson, Computer Associates
- 535 • John Hughes, Entegrity Solutions
- 536 • Carlisle Adams, Entrust
- 537 • Robert Griffin, Entrust
- 538 • Scott Cantor, Individual
- 539 • Bob Morgan, Individual
- 540 • Clifford Thompson, Individual
- 541 • Padraig Moloney, NASA
- 542 • Prateek Mishra, Netegrity (co-chair)
- 543 • Frederick Hirsch, Nokia
- 544 • Senthil Sengodan, Nokia
- 545 • Timo Skytta, Nokia
- 546 • Charles Knouse, Oblix
- 547 • Steve Anderson, OpenNetwork
- 548 • Simon Godik, OverXeer
- 549 • Rob Philpott, RSA Security (co-chair)
- 550 • Dipak Chopra, SAP
- 551 • Jahan Moreh, Sigaba
- 552 • Bhavna Bhatnagar, Sun Microsystems
- 553 • Jeff Hodges, Sun Microsystems
- 554 • Eve Maler, Sun Microsystems (coordinating editor)
- 555 • Emily Xu, Sun Microsystems
- 556 • Phillip Hallam-Baker, VeriSign

557

Appendix B. Notices

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

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

569 **Copyright © OASIS Open 2003. All Rights Reserved.**

570 This document and translations of it may be copied and furnished to others, and derivative works that
571 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
572 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
573 this paragraph are included on all such copies and derivative works. However, this document itself does
574 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as
575 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights
576 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it
577 into languages other than English.

578 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
579 or assigns.

580 This document and the information contained herein is provided on an "AS IS" basis and OASIS
581 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
582 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR
583 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

584

585

Appendix C. Revision History

Draft	Who	What
01	Eve Maler	Cosmetic changes to bring spec up to 1.1 WD status. Copyedits and editorial review. There are a few outstanding issues that the TC will need to address.
02	Eve Maler	Rationalized the “x assertion” wording. Took out the implication that the TC is providing an actual test suite.
03	Rob Philpott	Updated bibliography dates for all SAML specs. Accepted all changes in document for Last Call.

586