



SAML V2.0 Condition for Delegation Restriction

Committee Draft 01, 10 March 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation-cd-01.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation-cd-01.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation-cd-01.pdf>

Previous Version:

None

Latest Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation.pdf>

Latest Approved Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation-cd-01.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation-cd-01.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation-cd-01.pdf>

Technical Committee:

OASIS Security Services TC

Chair(s):

Hal Lockhart, BEA Systems, Inc.

Brian Campbell, Ping Identity Corporation

Editors:

Scott Cantor, Internet2

Abstract:

This document defines a `<saml:Condition>` type for expressing a chain of intermediaries acting on behalf of the subject of an assertion, requiring relying parties to distinguish between direct and indirect access.

Status

This document was last revised or approved by the SSTC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

35 TC members should send comments on this specification to the TC's email list. Others
36 should send comments to the TC by using the "Send A Comment" button on the TC's
37 web page at <http://www.oasis-open.org/committees/security>.
38 For information on whether any patents have been disclosed that may be essential to
39 implementing this specification, and any offers of patent licensing terms, please refer to the IPR
40 section of the TC web page (<http://www.oasis-open.org/committees/security/ipr.php>).
41 The non-normative errata page for this specification is located at [http://www.oasis-
open.org/committees/security](http://www.oasis-
42 open.org/committees/security).

43 Notices

44 Copyright © OASIS Open 2009. All Rights Reserved.

45 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
46 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

47 This document and translations of it may be copied and furnished to others, and derivative works that
48 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
49 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
50 and this section are included on all such copies and derivative works. However, this document itself may
51 not be modified in any way, including by removing the copyright notice or references to OASIS, except as
52 needed for the purpose of developing any document or deliverable produced by an OASIS Technical
53 Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be
54 followed) or as required to translate it into languages other than English.

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

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

62 OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
63 necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to
64 notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such
65 patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced
66 this specification.

67 OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any
68 patent claims that would necessarily be infringed by implementations of this specification by a patent
69 holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR
70 Mode of the OASIS Technical Committee that produced this specification. OASIS may include such
71 claims on its website, but disclaims any obligation to do so.

72 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
73 might be claimed to pertain to the implementation or use of the technology described in this document or
74 the extent to which any license under such rights might or might not be available; neither does it represent
75 that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to
76 rights in any document or deliverable produced by an OASIS Technical Committee can be found on the
77 OASIS website. Copies of claims of rights made available for publication and any assurances of licenses
78 to be made available, or the result of an attempt made to obtain a general license or permission for the
79 use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS
80 Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any
81 information or list of intellectual property rights will at any time be complete, or that any claims in such list
82 are, in fact, Essential Claims.

83 The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be
84 used only to refer to the organization and its official outputs. OASIS welcomes reference to, and
85 implementation and use of, specifications, while reserving the right to enforce its marks against
86 misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

87 **Table of Contents**

88 1 Introduction.....5
89 1.1 Notation.....5
90 1.2 Normative References.....6
91 1.3 Non-Normative References.....6
92 1.4 Conformance.....6
93 1.4.1 SAML V2.0 Condition for Delegation Restriction.....6
94 2 SAML V2.0 Condition for Delegation Restriction.....7
95 2.1 Required Information.....7
96 2.2 Overview.....7
97 2.3 Element <Delegate>.....7
98 2.4 Complex Type DelegationRestrictionType.....8
99 2.5 Use of Identifiers Within <saml:SubjectConfirmation>.....8
100 2.6 Security Considerations.....8
101 Appendix A. Acknowledgements.....9
102 Appendix B. Revision History.....10
103

1 Introduction

104

105 Some advanced SAML use cases involve a single logical transaction that spans one or more intermediate
106 clients or servers. An example includes a web site acting on behalf of a logged-in user while accessing a
107 third service. Generalizing this example, a number of intermediaries might be transited before the final
108 point of access. If a SAML assertion is used as a security token to authenticate and authorize such
109 access, it is important that the identity and order of intermediaries, if any, be expressed within the token in
110 some fashion.

111 Existing mechanisms designed for this purpose, such as the `<saml:SubjectConfirmation>` element
112 definition in the SAML V2.0 core specification [SAML2Core], or the extended syntax found in the Liberty
113 ID-WSF Security Mechanisms specification [LibSecMech20], suffer from the drawback that they have
114 advisory semantics for a relying party and are likely to be ignored by delegation-unaware SAML
115 processing. While backward compatibility can be an advantage, ignoring security-relevant details that
116 might impact upon a relying party's policy is unacceptable in some scenarios.

117 This specification provides for the expression of delegation information with normative SAML processing
118 semantics through the use of a `<saml:Condition>` extension type.

1.1 Notation

119

120 This specification uses normative text.

121 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
122 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as
123 described in [RFC2119]:

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

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

129 Listings of XML schemas appear like this.

130 Example code listings appear like this.

132 Conventional XML namespace prefixes are used throughout the listings in this specification to stand for
133 their respective namespaces as follows, whether or not a namespace declaration is present in the
134 example:

Prefix	XML Namespace	Comments
saml:	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace defined in the SAML V2.0 core specification [SAML2Core].
del:	urn:oasis:names:tc:SAML:2.0:conditions:delegation	This is the namespace defined by this specification.
xsd:	http://www.w3.org/2001/XMLSchema	This namespace is defined in the W3C XML Schema specification [Schema1]. In schema listings, this is the default namespace and no prefix is shown.
xsi:	http://www.w3.org/2001/XMLSchema-instance	This is the XML Schema namespace for schema-related markup that appears in XML instances [Schema1].

135 This specification uses the following typographical conventions in text: <SAML*E*lement>,
136 <ns:ForeignElement>, Attribute, **Datatype**, OtherCode.

137 1.2 Normative References

- 138 **[RFC2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
139 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 140 **[SAML2Core]** S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion*
141 *Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID
142 saml-core-2.0-os. See [http://docs.oasis-open.org/security/saml/v2.0/saml-core-](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)
143 [2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf).
- 144 **[Schema1]** H. S. Thompson et al. *XML Schema Part 1: Structures*. World Wide Web
145 Consortium Recommendation, May 2001. See [http://www.w3.org/TR/2001/REC-](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/)
146 [xmlschema-1-20010502/](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/). Note that this specification normatively references
147 [Schema2], listed below.
- 148 **[Schema2]** Paul V. Biron, Ashok Malhotra. *XML Schema Part 2: Datatypes*. World Wide Web
149 Consortium Recommendation, May 2001. See [http://www.w3.org/TR/2001/REC-](http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/)
150 [xmlschema-2-20010502/](http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/).

151 1.3 Non-Normative References

- 152 **[LibSecMech20]** F.Hirsch. *Liberty ID-WSF Security Mechanisms Core*. November 2006.
153 <http://www.projectliberty.org/specs>.

154 1.4 Conformance

155 1.4.1 SAML V2.0 Condition for Delegation Restriction

156 An assertion issuer conforms to this specification if it can generate assertions containing a
157 <saml:Condition> of type **DelegationRestrictionType**, per section 2.

158 A relying party conforms to this specification if it can successfully process assertions containing a
159 <saml:Condition> of type **DelegationRestrictionType**, per section 2.

2 SAML V2.0 Condition for Delegation Restriction

2.1 Required Information

Identification: urn:oasis:names:tc:SAML:2.0:conditions:delegation

Contact information: security-services-comment@lists.oasis-open.org

Description: Given below.

Updates: None.

2.2 Overview

The SAML V2.0 core specification [SAML2Core] defines the **saml:ConditionAbstractType** complex type as a basis for extensions with mandatory processing semantics for relying parties. This specification defines such an extension as a supplement for the presence of an identifier within the `<saml:SubjectConfirmation>` element.

Rather than an advisory mechanism for identifying a single delegate, the extension provides for a normative mechanism that identifies an ordered sequence of delegates, along with optional detail about the acts of delegation.

2.3 Element <Delegate>

The `<Delegate>` element is a container for a single intermediary/delegate represented by the assertion. It contains the following elements and attributes:

`DelegationInstant` [Optional]

A timestamp indicating the approximate time at which the act of delegation occurred, if known.

`ConfirmationMethod` [Optional]

Identifies the subject confirmation method used, if the delegate presented a SAML assertion to authenticate itself to the issuing authority.

`<saml:BaseID>`, `<saml:NameID>`, `<saml:EncryptedID>` [Required]

Identifies the delegate.

The delegate is identified by a required child element in the usual SAML fashion. The optional attributes, if present, supply additional information about the act of delegation.

The following schema fragment defines the `<Delegate>` element and its **DelegateType** complex type:

```
<element name="Delegate" type="del:DelegateType"/>
<complexType name="DelegateType">
  <choice>
    <element ref="saml:BaseID"/>
    <element ref="saml:NameID"/>
    <element ref="saml:EncryptedID"/>
  </choice>
  <attribute name="DelegationInstant" type="dateTime" use="optional"/>
  <attribute name="ConfirmationMethod" type="anyURI" use="optional"/>
</complexType>
```

197 2.4 Complex Type DelegationRestrictionType

198 The **DelegationRestrictionType** complex type defines a subtype of **saml:ConditionType** representing
199 one or more acts of delegation that are represented by the containing assertion. It contains the following
200 elements:

201 <Delegate> [One or more]

202 An element identifying a delegate of the subject of the containing assertion. The delegates MUST be
203 ordered from least to most recent; thus the earliest element is the farthest removed from the
204 immediate use of the assertion.

205 A relying party MUST evaluate the list of delegates, and SHOULD NOT accept the assertion unless it
206 wishes to permit each delegate to act on behalf of the subject of the containing assertion.

207 A SAML authority MUST NOT include more than one <saml:Condition> element of this type within a
208 <saml:Conditions> element of an assertion.

209 For the purposes of determining the validity of the <saml:Conditions> element, this condition type is
210 always considered to be valid. That is, this condition type does not affect assertion validity, but is a
211 condition on use.

212 The following schema fragment defines the **DelegationRestrictionType** complex type:

```
213 <complexType name="DelegationRestrictionType">  
214   <complexContent>  
215     <extension base="saml:ConditionAbstractType">  
216       <sequence>  
217         <element ref="del:Delegate" maxOccurs="unbounded"/>  
218       </sequence>  
219     </extension>  
220   </complexContent>  
221 </complexType>
```

222 2.5 Use of Identifiers Within <saml:SubjectConfirmation>

223 For consistency with the existing SAML-defined syntax, it is RECOMMENDED that the identifier of the
224 most recent delegate (within the last element in the condition, per section 2.4) be duplicated within the
225 relevant <saml:SubjectConfirmation> elements in the containing assertion.

226 2.6 Security Considerations

227 The content of this condition type is directly impacted by the security semantics of the flow of activity that
228 leads to the issuance of the containing assertion. This specification does not define the exchanges that
229 must take place, and relies on composition with other profiles that logically represent acts of delegation
230 that require representation in an assertion.

231 Relying parties are not required to apply any particular policies with regard to the information represented
232 by this condition type. Rather, it is expected that such information will naturally be significant in the
233 enforcement of existing policies, and that the presence of delegation is significant enough to warrant the
234 disruption of existing services designed to consume SAML assertions until those policies reflect a
235 willingness to accept more indirect forms of access.

236 **Appendix A. Acknowledgements**

237 The editors would like to acknowledge the contributions of the OASIS Security Services Technical
238 Committee, whose voting members at the time of publication were:

- 239 • Rob Philpott, EMC Corporation
- 240 • John Bradley, Individual
- 241 • Jeff Hodges, Individual
- 242 • Scott Cantor, Internet2
- 243 • Nate Klingenstein, Internet2
- 244 • Bob Morgan, Internet2
- 245 • Joni Brennan, Liberty Alliance Project
- 246 • Tom Scavo, National Center for Supercomputing Applications (NCSA)
- 247 • Frederick Hirsch, Nokia Corporation
- 248 • Ari Kermaier, Oracle Corporation
- 249 • Hal Lockhart, Oracle Corporation
- 250 • Brian Campbell, Ping Identity Corporation
- 251 • Anil Saldhana, Red Hat
- 252 • Kent Spaulding, Skyworth TTG Holdings Limited
- 253 • Emily Xu, Sun Microsystems
- 254 • Duane DeCouteau, Veterans Health Administration
- 255 • David Staggs, Veterans Health Administration

256 **Appendix B. Revision History**

- 257 ● Draft 01
- 258 ● Committee Draft 01, CD edits