MSIS

Web Services Security: SAML Token Profile

4 Working Draft 06, 21 February 2003

5 6	Document identifier: WSS-SAML-06			
7 8	Location: TBD			
9 0 1 2 3	Editors: Phillip Hallam-Baker, VeriSign Chris Kaler, Microsoft Ronald Monzillo, Sun Anthony Nadalin, IBM			
4	Contributors:			
15	TBD - Revise this list to include WSS TC contributors			
	Phillip Hallam-Baker, VeriSign Jeff Hodges, Sun Microsystems Maryann Hondo, IBM Chris Kaler, Microsoft Eve Maler, Sun Microsystems Hiroshi Maruyama, IBM Chris McLaren, Netegrity Prateek Mishra, Netegrity Anthony Nadalin, IBM Nataraj Nagaratnam, IBM Hemma Prafullchandra, VeriSign Irving Reid, Baltimore Krishna Sankar, Cisco John Shewchuk, Microsoft			
6 7 8	Abstract: This document describes how to use Security Assertion Markup Language (SAML) assertions with the WS-Security specification.			
19 20 21	Status: This is an interim draft. Please send comments to the editors.			
22 23 24 25	Committee members should send comments on this specification to wss@lists.oasis-open.org list. Others should subscribe to and send comment to the wss-comment@lists.oasis-open.org list. To subscribe, visit http://lists.oasis-open.org/ob/adm.pl.			

For information on the disclosure of Intellectual Property Rights or licensing terms related to the work of the Web Services Security TC please refer to the Intellectual Property Rights section of the TC web page at http://www.oasis-open.org/committees/wss/. The OASIS policy on Intellectual Property Rights is described at http://www.oasis-open.org/who/intellectualproperty.shtml.

26 27

28 29

30

31

WSS-SAML-02

Copyright © OASIS Open 2002. All Rights Reserved.

Table of Contents

32	1 Introduction	4
33	1.1 Goals and Requirements	4
34	1.1.1 Requirements	4
35	1.1.2 Non-Goals	4
36	2 Notations and Terminology	5
37	2.1 Notational Conventions	5
38	2.2 Namespaces	5
39	2.3 Terminology	6
40	3 Usage	7
41	3.1 Processing Model	7
42	3.2 Attaching Security Tokens	7
43	3.3 Identifying and Referencing Security Tokens	8
44	3.4 Proof-of-Possession of Security Tokens	10
45	3.5 Error Codes	12
46	3.6 Threat Model and Countermeasures	18
47	4 Acknowledgements	21
48	5 References	22
49	Appendix A: Revision History	24
50	Appendix B: Notices	25
51		

1 Introduction

- 53 The WS-Security specification proposes a standard set of SOAP extensions that can
- 54 be used when building secure Web services to implement message level integrity and
- 55 confidentiality. This specification describes the use of Security Assertion Markup
- 56 Language (SAML) assertions from the <wsse:Security> header block defined by the
- 57 WS-Security specification.

1.1 Goals and Requirements

- 59 The goal of this specification is to define the use of SAML assertions in the context of
- 60 WS-Security including for the purpose of securing SOAP message exchanges.
- The requirements to be satisfied by this specification are listed below.
- 62 1.1.1 Requirements
- 63 TBS

52

58

- 64 1.1.2 Non-Goals
- The following topics are outside the scope of this document:
- 66 TBS

67

2 Notations and Terminology

This section specifies the notations, namespaces, and terminology used in this specification.

71 2.1 Notational Conventions

- 72 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
- 73 "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
- document are to be interpreted as described in RFC2119.
- Namespace URIs (of the general form "some-URI") represent some application-
- dependent or context-dependent URI as defined in RFC2396.
- 77 This specification is designed to work with the general SOAP message structure and
- 78 message processing model, and should be applicable to any version of SOAP. The
- 79 current SOAP 1.2 namespace URI is used herein to provide detailed examples, but
- 80 there is no intention to limit the applicability of this specification to a single version
- 81 of SOAP.

68

- 82 Readers are presumed to be familiar with the terms in the Internet Security
- 83 Glossary.

2.2 Namespaces

The XML namespace URIs that MUST be used by implementations of this specification are as follows (note that different elements in this specification are from different namespaces):

```
http://schemas.xmlsoap.org/ws/2002/xx/secext
http://schemas.xmlsoap.org/ws/2002/xx/utility
```

The following namespaces are used in this document:

90 91

84

85

86

87 88

89

Prefix	Namespace	
S http://www.w3.org/2001/12/soap-envelop		
ds	http://www.w3.org/2000/09/xmldsig#	
xenc	http://www.w3.org/2001/04/xmlenc#	
wsse	http://schemas.xmlsoap.org/ws/2002/xx/secext	
wsu	http://schemas.xmlsoap.org/ws/2002/xx/utility	

saml	urn:	oasis:names:tc:SAML:1.0:assertion
samlp	urn:	oasis:names:tc:SAML:1.0:protocol

92 2.3 Terminology

- 93 This specification employs the terminology defined in the WS-Security Core
- 94 Specification.
- 95 Defined below are the basic definitions for additional terminology used in this
- 96 specification.
- 97 Sender
- 98 Subject

3 Usage

- 100 This section describes the specific mechanisms and procedures for the SAML profile
- 101 of WS-Security.

99

107

123

- 102 Identification: urn:oasis:names:tc:WSS:1.0:profiles:WSS-SAML-profile
- 104 Contact information: TBD
- 105 **Description:** Given below.
- 106 Updates: None.

3.1 Processing Model

- 108 The SAML profile of WS-Security extends the token-independent processing model
- defined by the core WS-Security specification.
- 110 When a receiver processes a <wsse:Security> header containing or referencing
- 111 SAML assertions, it MUST select, based on its policy, the signatures and assertions
- that it will process. It is assumed that a receiver's signature selection policy may rely
- on semantic labeling of <wsse:SecurityTokenReference> elements occurring in the
- 114 <ds:KeyInfo> elements within the signatures. It is also assumed that the assertions
- 115 selected for validation and processing will include those referenced from the
- 116 <ds:KeyInfo> and <ds:SignedInfo> elements of the selected signatures.
- 117 As part of its validation and processing of the selected assertions, the receiver MUST
- 118 make an explicit determination of the relationship between the subject of each
- 119 assertion and the sender of the message. Two methods for establishing this
- 120 correspondence, holder-of-key and sender-vouches are described below. Senders
- 121 and receivers implementing the SAML profile of WS-Security MUST implement the
- 122 processing necessary to support both of these subject confirmation methods.

3.2 Attaching Security Tokens

- 124 SAML assertions are attached to SOAP messages using WS-Security by placing
- assertion elements or references to assertions inside a <wsse:Security>header.
- The following example illustrates a SOAP message containing a SAML assertion in a
- 127 <wsse:Security> header.

```
128
           <S:Envelope xmlns:S="...">
129
               <S:Header>
130
                   <wsse:Security xmlns:wsse="...">
131
                       <saml:Assertion</pre>
132
                                 MajorVersion="1"
133
                                  MinorVersion="0"
134
                                  AssertionID="SecurityToken-ef375268"
135
                                  Issuer="elliotw1"
136
                                  IssueInstant="2002-07-23T11:32:05.6228146-07:00"
```

WSS-SAML-02 Copyright © OASIS Open 2002. All Rights Reserved. 23 September 2002

```
xmlns.saml="urn.oasis.names.tc.SAMI.1 0.assertion">
137
139
                        </saml:Assertion>
140
141
                   </wsse:Security>
142
               </S:Header>
143
               <S:Body>
144
                   . . .
145
               </S:Body>
146
           </S:Envelope>
```

3.3 Identifying and Referencing Security Tokens

147

148

149

150

151

152

153

154

155

156

157158

159

160 161

162

163

164

165

166

167

168 169

- The WS-Security specification defines the <wsse:SecurityTokenReference> element for referencing security tokens. Three forms of token references are defined by this element and the element schema includes provision for defining additional reference forms should they be necessary. The three forms of token references defined by the < wsse:SecurityTokenReference> element are defined as follows:
- A key identifier reference a generic element (i.e. < wsse: Keyldentifier>) that
 conveys a security token identifier and indicates in its attributes (as necessary)
 the type of the token being identified (i.e. the ValueType), the identifier encoding
 type (i.e. the EncodingType), and any other parameters necessary to reference
 the security token.
 - When a key identifier is used to reference a SAML assertion the ValueType attribute must contain the value "saml: Assertion" and the < wsse: Keyldentifier> element must contain as its element value the corresponding AssertionID.
 - The SAML profile of WSS-Security prescribes the use of the following attributes within a key identifier reference when the referenced assertion must be acquired from the assertion authority.
 - /wsse: Security Token Reference/Key I dentifier/@sam1: Location
 - This optional attribute is used to carry a URI reference describing how to locate the SAML authority. As defined by SAMLCore, the syntax of the URI will depend on the protocol binding defined by the saml: Binding attribute of the < wsse: Keyldentifier>. For example, a binding based on HTTP will be a web URL, while a binding based on SMTP might use the "mailto" scheme.
- 170 /wsse:SecurityTokenReference/keyIdentifier/@saml:Binding
- 171 A URI reference identifying the SAML protocol binding to use in 172 communicating with the SAML authority. SAML protocol bindings are assigned 173 a URI reference in SAMLBind.
- 174 { Note to TC: this mechanism should be extended to support artifact references}
- A key name reference a <ds:KeyName> element contains a string value key
 identifier, and the referenced token or tokens are those that contain a matching
 identity value.

- 179 The syntax of SAML assertion identifiers does not facilitate their differentiation 180 from other identifier forms. For this reason, key name reference forms SHOULD 181 not be used to reference SAML assertions.
- 182 A Direct or URI reference – a generic element (i.e. < wsse: Reference>) that identifies a security token by URI. If only a fragment is specified, then the 183 184 reference is to the security token within the document whose wsu:Id attribute value matches the fragment. Otherwise, the reference is to the (potentially 185 186 external) security token identified by the URI.
- 187 The SAML assertion schema does not include or provide for inclusion of the 188 wsu:Id attribute. For this reason, a URI reference cannot be used to (directly) reference a SAML assertion. 189
- 190 In the SAML profile of WS-security, SAML assertions may be referenced in three 191 contexts:
- 192 A SAML assertion may be referenced from a < ds: KeyInfo> element of a 193 <ds:Signature> element in a < wsse:Security> header. In this case, the assertion contains the key used in the signature calculation. 194
- 195 A SAML assertion may be referenced from a < ds: Reference> element within the 196 <ds:SignedInfo> element of a < ds:Signature> element in a < wsse:Security> 197 header. In this case, the referenced assertion is being signed by the containing 198 signature.
- 199 A SAML assertion may be referenced from a < wsse: Security > header or from an 200 element (other than a signature) in the header.
- 201 In each of these contexts, the referenced assertion may be:
- 202 local - in which case, it is included in the <wsse:Security> header containing the 203 reference.
- 204 remote – in which case it is not included in the <wsse:Security> header 205 containing the reference, but may occur in another part of the SOAP message or 206 may be available at the location identified by the reference which may be an 207 assertion authority.
- 208 In the SAML profile of WS-Security, the preferred method to reference SAML assertions is by key identifier reference. 209
- A SAML assertion that exists in a < wsse: Security> header may be referenced from 210
- 211 the < wsse: Security> header, a header element, or from the < ds: KeyInfo> element
- 212 of a < ds: Signature> element in the header by using a key identifier reference.
- 213 Methods to reference SAML assertion from a < ds: Reference> element remain to be 214 formalized.

3.3.1 SAML Assertion Referenced from Header or Element 215

216 A SAML assertion may be referenced from a < wsse: Security > header or from an 217 element (other than a signature) in the header. The following example demonstrates

WSS-SAML-02 23 September 2002 the use of a key identifier reference in a < wsse: Security> header to reference a local SAML assertion.

218

219

244

245

246

247

256

258

259

260

261

262

```
220
           <S:Envelope xmlns:S="...">
221
               <S:Header>
222
                   <wsse:Security xmlns:wsse="...">
223
                        <saml:Assertion
224
                                  MajorVersion="1"
225
                                  MinorVersion="0"
226
                                  AssertionID="SecurityToken-ef375268"
227
                                  Issuer="elliotw1"
228
                                  IssueInstant="2002-07-23T11:32:05.6228146-07:00"
229
                                xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
230
                            . . .
231
                        </saml:Assertion>
232
                       <wsse:SecurityTokenReference</pre>
233
                            <wsse:KeyIdentifier wsu:id="..."</pre>
234
                                   ValueType="saml:Assertion"
235
                                   SecurityToken-ef375268
236
                            </wsse:KeyIdentifier>
237
                       </wsse:SecurityTokenReference>
238
                    </wsse:Security>
239
               </S:Header>
240
               <S:Body>
241
242
               </S:Body>
243
           </S:Envelope>
```

A SAML assertion that exists outside of a < wsse: Security> header may be referenced from the < wsse: Security> header element by including (in the reference) saml: Location and saml: Binding attributes that define the address and protocol to use to acquire the identified assertion at a SAML assertion authority or responder.

```
248
            <wsse:SecurityTokenReference</pre>
249
               <wsse:KeyIdentifier wsu:id="..."</pre>
250
                   ValueType="saml:Assertion"
251
                   saml:Location=http://www.fabrikam123.com/elliotw1
252
                   saml:Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
253
                   SecurityToken-ef375268
254
               </wsse:KeyIdentifier>
255
            </wsse:SecurityTokenReference>
```

3.3.2 SAML assertion referenced from KeyInfo

The following examples demonstrate the use of a key identifier reference from within a < ds: KeyInfo> element of a < ds: Signature> element in a < wsse: Security> header.

The following example depicts the use of a key identifier reference containing a SAML AssertionID (as its value) to reference a local assertion identified by AssertionID. { It is presumed that the default encoding type is xsi:string}.

```
269
270 </wsse-SecurityTokenReference>
</ds:KeyInfo>
```

271

272

273

284

290

298

299

300

301

The following example extends the previous example with the inclusion of saml: Location and saml: Binding attributes that define the address and protocol to use to acquire the identified assertion at a SAML assertion authority or responder.

```
274
           <ds:KeyInfo>
275
              <wsse:SecurityTokenReference>
276
                  <wsse:KeyIdentifier wsu:id="..."</pre>
277
                     ValueType="saml:Assertion"
278
                     saml:Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
279
                     saml:Location="http://www.fabrikam123.com/elliotw1"
280
                     SecurityToken-ef375268
281
                  </wsse:KeyIdentifier>
282
              </wsse:SecurityTokenReference>
283
           </ds:KeyInfo>
```

3.3.3 SAML assertion referenced from SignedInfo

286 { Note to TC: Methods to reference SAML assertions from < ds: Reference> elements
287 remain to be formalized. One issue that remains to be resolved is how to
288 differentiate whether it is the reference or the referenced assertion that is to be
289 digested.}

3.4 Subject Confirmation of SAML Assertions

The SAML profile of WS-Security requires that message senders and receivers support the holder-of-key and sender-vouches methods of subject confirmation. It is strongly RECOMMENDED that an XML signature be used to establish the relationship between the message sender and the attached assertions. This is especially RECOMMENDED whenever the SOAP message exchange is conducted over an unprotected transport.

Any processor of SAML assertions MUST conform to the required validation and processing rules defined in the SAML specification.

The following table enumerates the mandatory subject confirmation methods and summarizes their associated processing models:

Mechanism	RECOMMENDED Processing Rules
<pre>urn:oasis:names:tc:SAML:1.0:cm:holder- of-key</pre>	The requestor includes an XML Signature that can be verified with the key information in the < saml: ConfimationMethod> of the SAML assertion referenced by the Signature.

Urn:oasis:names:tc:SAML:1.0:cm:sendervouches

The requestor (the sender, different from the subject) vouches for the verification of the subject. The receiver MUST have an existing trust relationship with the requestor to accept this. It is RECOMMENDED that the requestor sign the token and the message or use a secure transport.

Note that the high level processing model described in the following sections does not differentiate between message author and message sender as would be necessary to guard against replay attacks. The high-level processing model also does not take into account requirements for authentication of receiver by sender, or for message or assertion confidentiality. These concerns must be addressed by means other than those described in the high-level processing model.

3.4.1 Holder-of-key Subject Confirmation Method

The following sections describe the holder-of-key method of establishing the correspondence between a SOAP message sender and the subject of SAML assertions added to the SOAP message according to the SAML profile of WS-Security.

312 3.4.1.1 Sender

302

303

304

305

306

307

308

309

310

311

314

315

316

326

327

328

329

330

331

313 A message sender uses the holder-of-key confirmation method to demonstrate that it is authorized to act as the subject of the assertions in the message. The assertions included in a message that the sender will confirm by the holder-of-key method MUST include the following <saml:SubjectConfirmation> element:

```
317
           <saml:SubjectConfirmation>
318
             <saml:ConfirmationMethod>
319
               urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
320
             </saml:ConfirmationMethod>
321
             <ds:KeyInfo>...</ds:KeyInfo>
322
           </saml:SubjectConfirmation>
```

The <saml:SubjectConfirmation> element MUST include a <ds:KeyInfo> element 323 324 that identifies the public or secret key to be used to confirm the identity of the 325 subject.

To satisfy the associated confirmation method processing of the message receiver. the sender MUST demonstrate knowledge of the confirmation key. The sender MAY accomplish this by using the confirmation key to sign content within the message and by including the resulting <ds:Signature> element in the <wsse:Security> header.

<ds:Signature> elements produced for this purpose MUST conform to the 332 canonicalization and token inclusion rules defined in the core WS-Security 333 specification.

- 334 SAML assertions that contain a holder-of-key <saml:SubjectConfirmation> element
- 335 SHOULD contain a <ds:Signature> element that protects the integrity of the
- 336 confirmation <ds:KeyInfo> established by the assertion authority.
- 337 The canonicalization method used to produce the <ds:Signature> elements used
- 338 to protect the integrity of SAML assertions MUST support the validation of these
- 339 <ds:Signature> elements in contexts (such as < wsse:Security> header elements)
- other than those in which the signatures were calculated.

3.4.1.2 Receiver

341

351

352

353

354

- 342 Of the SAML assertions it selects for processing, a message receiver MUST NOT
- 343 accept assertions containing a holder-of-key <saml:ConfirmationMethod>, unless
- 344 the receiver has validated the integrity of the assertions and the message sender has
- 345 demonstrated knowledge of the key identified by the <ds:keyInfo> element of the
- 346 <saml:SubjectConfirmation> element. If the receiver determines that the sender
- 347 has demonstrated knowledge of a subject confirmation key, then the SAML
- 348 assertions containing the confirmation key MAY be attributed to the sender and any
- 349 elements of the message whose integrity is protected by the subject confirmation
- 350 key MAY be considered to have been authored by the subject.

3.4.1.3 Example

The following example illustrates the use of the holder-of-key subject confirmation method to establish the correspondence between the SOAP message author and the subject of the SAML assertions in the <wsse:Security> header:

```
355
           <?xml:version="1.0" encoding="UTF-8"?>
356
357
           <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
358
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
359
             xmlns:xsd="http://www.w3.org/2001/XMLSchema">
360
361
           <S:Header>
362
           <wsse:Security>
363
364
             <saml:Assertion
365
              xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
366
              MajorVersion="1" MinorVersion="0"
367
              AssertionID="2sxJu9q/vvLG9sAN9bKp/8q0NKU="
368
             Issuer="www.example.com"
369
              IssueInstant="2002-06-19T16:58:33.173Z">
370
               <saml:Conditions</pre>
371
                NotBefore="2002-06-19T16:53:33.173Z"
372
                NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
373
374
              <saml:AuthenticationStatement</pre>
375
                AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
376
                AuthenticationInstant="2002-06-19T16:57:30.000Z">
377
                <saml:Subject>
378
                  <saml:NameIdentifier</pre>
379
                    NameQualifier="www.example.com"
380
                    Format="">
381
                           uid=joe,ou=people,ou=saml-demo,o=example.com
382
                   </saml:NameIdentifier>
```

```
383
                   <saml:SubjectConfirmation>
384
                     <saml:ConfirmationMethod>
385
                           urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
386
                     </saml:ConfirmationMethod>
387
                     <ds:KeyInfo>
388
                       <ds:KeyValue>...</ds:KeyValue>
389
                     </ds:KeyInfo>
390
                   </saml:SubjectConfirmation>
391
                 </saml:Subject>
392
               </saml:AuthenticationStatement>
393
394
               <saml:AttributeStatement>
395
                 <saml:Subject>
396
                   <saml:NameIdentifier</pre>
397
                     NameQualifier="www.example.com"
398
                     Format="">
399
                           uid=joe, ou=people, ou=saml-demo, o=baltimore.com
400
                   </saml:NameIdentifier>
401
                   <saml:SubjectConfirmation>
402
                     <saml:ConfirmationMethod>
403
                            urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
404
                     </saml:ConfirmationMethod>
405
                     <ds:KeyInfo>
406
                       <ds:KeyValue>...</ds:KeyValue>
407
                     </ds:KeyInfo>
408
                   </saml:SubjectConfirmation>
409
                 </saml:Subject>
410
411
                 <saml:Attribute</pre>
412
                   AttributeName="MemberLevel"
413
                   AttributeNamespace="http://www.oasis-
414
           open.org/Catalyst2002/attributes">
415
                     <saml:AttributeValue>gold</saml:AttributeValue>
416
                 </saml:Attribute>
417
                 <saml:Attribute</pre>
418
                   AttributeName="E-mail"
419
                   AttributeNamespace="http://www.oasis-
420
           open.org/Catalyst2002/attributes">
421
                   <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
422
                 </saml:Attribute>
423
               </saml:AttributeStatement>
424
               <ds:Signature>...</ds:Signature>
425
             </saml:Assertion>
426
427
             <ds:Signature>
428
               <ds:SignedInfo>
429
                 <ds:CanonicalizationMethod Algorithm=</pre>
430
                   "http://www.w3.org/2001/10/xml-exc-c14n#"/>
431
                 <ds:SignatureMethod Algorithm=
432
                  "http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
433
                 </ds:Reference>
434
                 <ds:Reference URI="#MsgBody">
435
                   <ds:DigestMethod Algorithm=
436
                      "http://www.w3.org/2000/09/xmldsig#sha1"/>
437
                   <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
438
                 </ds:Reference>
439
              </ds:SignedInfo>
440
               <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
441
               <ds:KevInfo>
442
                 <wsse:SecurityTokenReference>
```

```
443
                    <wsse:Keyidentifier ValueType=saml:Assertion</pre>
444
           2sxJu9q/vvLG9sAN9bKp/8q0NKU=
445
                   </wsse:Keyidentifier >
446
                </wsse:SecurityTokenReference>
447
               </ds:KeyInfo>
448
            </ds:Signature>
449
450
           </wsse:Security>
451
           </S:Header>
452
453
           <S:Body wsu:Id="MsgBody">
454
            <ReportRequest>
455
              <TickerSymbol>SUNW</TickerSymbol>
456
            </ReportRequest>
457
           </S:Body>
458
           </S:Envelope>
```

3.4.2 Sender-vouches Subject Confirmation Method

- The following sections describe the sender-vouches method of establishing the 460 correspondence between a SOAP message sender and the SAML assertions added to 461
- 462 the SOAP message according to the SAML profile of WS-Security.

463 3.4.2.1 Sender

459

473

474

475

476

477

478

464 A message sender uses the sender-vouches confirmation method to assert that it is 465 acting on behalf of the subjects of the assertions in the message. The assertions 466 included in a message that the sender will confirm by the sender-vouches method 467 MUST include the following <saml:SubjectConfirmation> element:

```
468
           <saml:SubjectConfirmation>
469
            <saml:ConfirmationMethod>
470
                    urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
471
            </saml:ConfirmationMethod>
472
          </saml:SubjectConfirmation>
```

To satisfy the associated confirmation method processing of the receiver, the sender MUST integrity protect the assertions and those elements of the SOAP message that it is vouching for. The sender MAY accomplish this by including in the corresponding <wsse:Security> header a <ds:Signature> element that the sender prepares by using its key to sign the assertions and relevant message content. As defined by the XML Signature Specification, the sender MAY identify its key by including a <ds:KeyInfo> element within the <ds:Signature> element.

479

480 A <ds:Signature> element produced for this purpose MUST conform to the canonicalization and token inclusion rules defined in the core WS-Security 481

482 specification.

3.4.2.2 **Receiver** 483

484 Of the SAML assertions it selects for processing, a message receiver MUST NOT

485 accept assertions containing a sender-vouches <saml:ConfirmationMethod> unless

486 the assertions and SOAP message content being vouched for by the sender are

integrity protected by a sender who is trusted by the receiver to act on behalf of the subject of the assertions.

3.4.2.3 Example

487

488

489

490

491

492

493

The following example illustrates a sender's use of the sender-vouches subject confirmation method with an associated <ds:Signature> element to establish its identity and to assert that it has sent message elements on behalf of the subjects of the contained assertions:

```
494
           <?xml:version="1.0" encoding="UTF-8"?>
495
           <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
496
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
497
             xmlns:xsd="http://www.w3.org/2001/XMLSchema">
498
499
           <S:Header>
500
           <wsse:Security>
501
502
             <saml:Assertion</pre>
503
              xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
504
             MajorVersion="1" MinorVersion="0"
505
              AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
506
               Issuer="www.example.com"
507
               IssueInstant="2002-06-19T16:58:33.173Z">
508
               <saml:Conditions</pre>
509
                 NotBefore="2002-06-19T16:53:33.173Z"
510
                 NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
511
512
               <saml:AuthenticationStatement</pre>
513
                 AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
514
                 AuthenticationInstant="2002-06-19T16:57:30.000Z">
515
                 <saml:Subject>
516
                   <saml:NameIdentifier</pre>
517
                    NameQualifier="www.example.com"
518
                    Format="">
519
                           uid=joe, ou=people, ou=saml-demo, o=example.com
520
                   </saml:NameIdentifier>
521
                   <saml:SubjectConfirmation>
522
                     <saml:ConfirmationMethod>
523
                           urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
524
                     </saml:ConfirmationMethod>
525
                   </saml:SubjectConfirmation>
526
                 </saml:Subject>
527
               </saml:AuthenticationStatement>
528
529
               <saml:AttributeStatement>
530
                 <saml:Subject>
531
                   <saml:NameIdentifier</pre>
532
                    NameQualifier="www.example.com"
533
                     Format="">
534
                           uid=joe,ou=people,ou=saml-demo,o=baltimore.com
535
                   </saml:NameIdentifier>
536
                   <saml:SubjectConfirmation>
537
                     <saml:ConfirmationMethod>
538
                           urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
539
                     </saml:ConfirmationMethod>
540
                   </saml:SubjectConfirmation>
541
                 </saml:Subject>
```

```
542
543
                 <saml:Attribute
544
                   AttributeName="MemberLevel"
545
                   AttributeNamespace="http://www.oasis-
546
           open.org/Catalyst2002/attributes">
547
                     <saml:AttributeValue>gold</saml:AttributeValue>
548
                 </saml:Attribute>
549
                 <saml:Attribute</pre>
550
                   AttributeName="E-mail"
551
                   AttributeNamespace="http://www.oasis-
552
           open.org/Catalyst2002/attributes">
553
                   <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
554
                 </saml:Attribute>
555
               </saml:AttributeStatement>
556
             </saml:Assertion>
557
558
             <ds:Signature>
559
               <ds:SignedInfo>
560
                 <ds:CanonicalizationMethod Algorithm=</pre>
561
                   "http://www.w3.org/2001/10/xml-exc-c14n#"/>
562
                 <ds:SignatureMethod Algorithm=
563
                  "http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
564
                 <ds:Reference URI=#2sxJu9g/vvLG9sAN9bKp/8q0NKU=</pre>
565
                                Type= "saml:IDReferenceType">
566
                   <ds:DigestMethod Algorithm=
567
                     "http://www.w3.org/2000/09/xmldsig#sha1"/>
568
                   <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
569
                 </ds:Reference>
570
                 <ds:Reference URI="#MsgBody">
571
                   <ds:DigestMethod Algorithm=
572
                     "http://www.w3.org/2000/09/xmldsig#sha1"/>
573
                   <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
574
                 </ds:Reference>
575
               </ds:SignedInfo>
576
               <ds:SignatureValue>JWbvqW94vJVQkA...</ds:SignatureValue>
577
               <ds:KeyInfo>
578
                <X509Data>
579
                   <X509SubjectName>portal@yahoo.com</X509SubjectName>
580
                 </X509Data>
581
               </ds:KeyInfo>
582
             </ds:Signature>
583
584
           </wsse:Security>
585
           </S:Header>
586
587
           <S:Body wsu:Id="MsgBody">
588
             <ReportRequest>
589
               <TickerSymbol>SUNW</TickerSymbol>
590
             </ReportRequest>
591
           </S:Body>
592
593
           </S:Envelope>
```

3.5 Error Codes

594

It is RECOMMENDED that systems that implement the SAML profile of WS-Security respond with the error codes defined in the core WS-Security specification.

Implementations that chose to respond with custom errors, defined in private

WSS-SAML-02

namespaces, SHOULD take care not to introduce any security vulnerabilities as a result of the information returned in their error responses.

A receiver that is unable to process the SAML assertions contained in or referenced from a <wsse:Security> header MUST use one of the fault codes listed in the core WS-Security specification to report the error. The RECOMMENDED correspondence between the common assertion processing failures and the error codes defined in the core WS-security specification are defined in the following table:

Assertion Processing Error	RECOMMENDED Error
A referenced SAML assertion could not be retrieved.	Wsse:SecurityTokenUnavailable
An assertion contains a <saml:condition> element that the receiver does not understand.</saml:condition>	Wsse:UnsupportedSecurityToken
A signature within an assertion or referencing an assertion is invalid.	Wsse:FailedCheck
The issuer of an assertion is not acceptable to the receiver.	Wsse:InvalidSecurityToken
The receiver does not understand the extension schema used in an assertion.	Wsse:UnsupportedSecurityToken

3.6 Threat Model and Countermeasures

- This document defines the mechanisms and procedures for securely attaching SAML
- assertions to SOAP messages. SOAP messages are used in multiple contexts,
- specifically including cases where the message is transported without an active
- session, the message is persisted, or the message is routed through a number of
- intermediaries. Such a general context of use suggests that users of this profile must be concerned with a variety of threats. The following sections describe the
- vulnerability of the SAML token profile of WS-Security. In general, the use of SAML
- assertions with WS-Security introduces no new threats beyond those identified for
- 614 SAML or by the core WS-Security specification.
- The following sections provide an overview of the characteristics of the threat model,
- and the countermeasures that SHOULD be adopted for each perceived threat.

3.6.1 Eavesdropping

598

599

600

601

602

603

604

605

617

- Eavesdropping is a threat to the SAML token profile of WS-Security in the same
- 619 manner as it is a threat to any network protocol. The routing of SOAP messages
- through intermediaries increases the potential incidences of eavesdropping.
- 621 Additional opportunities for eavesdropping exist when SOAP messages are persisted.

622 623 624 625 626	To provide maximum protection from eavesdropping, assertions, assertion references, and sensitive message content SHOULD be encrypted such that only the intended audiences can view their content. This removes threats of eavesdropping in transit, but MAY not remove risks associated with storage or poor handling by the receiver.
627 628 629 630	Transport-layer security MAY be used to protect the message and contained SAML assertions and/or references from eavesdropping while in transport, but message content MUST be encrypted above the transport if it is to be protected from eavesdropping by intermediaries.

3.6.2 Replay

631

650

- The reliance on authority protected (e.g. signed) assertions with a holder-of-key 632 633 subject confirmation mechanism precludes all but a holder of the key from binding 634 the assertions to a SOAP message. Although this mechanism affectively restricts message authorship to the holder of the confirmation key, it does not preclude the 635 636 capture and resubmission of the message by other parties.
- 637 Assertions that contain a sender-vouches confirmation mechanism introduce another 638 dimension to replay vulnerability because the assertions impose no restriction on the 639 senders who may use or reuse the assertions. Any entity coming into contact with 640 such assertions could use them in a message in which they use their identity to 641 vouch for the subject of the assertions.
- 642 Replay attacks can be addressed by using message timestamps and caching, as well 643 as by using other application-specific tracking mechanisms.

644 3.6.3 Message Insertion

645 The SAML token profile of WS-Security is not vulnerable to message insertion 646 attacks.

647 3.6.4 Message Deletion

648 The SAML token profile of WS-Security is not vulnerable to message deletion attacks.

3.6.5 Message Modification

651 The SAML token profile of WS-Security is protected from message modification if the 652 relevant message content is integrity protected by the holder of the key or by the 653 vouching sender. Therefore, it is strongly RECOMMENDED that all relevant and 654 immutable message content be signed by the holder of the key or by the vouching sender (as the case warrants). Receivers SHOULD only consider those portions of the 655 656 document that are integrity protected by the appropriate entity as being subject to 657 the assertions in the message.

WSS-SAML-02 23 September 2002 Copyright © OASIS Open 2002. All Rights Reserved.

659 660 661 662 663 664 665	To ensure that message receivers can have confidence that received assertions have not been forged or altered since their issuance, SAML assertions and assertion references appearing in <wsse:security> header elements MUST be integrity protected (e.g. signed) by their issuing authority or the vouching sender (as the case warrants). It is strongly RECOMMENDED that a message sender sign any <saml:assertion> elements that it is confirming and that are not signed by their issuing authority.</saml:assertion></wsse:security>
667 668 669	Transport-layer security MAY be used to protect the message and contained SAML assertions and/or assertion references from modification while in transport, but signatures are required to extend such protection through intermediaries.
670	3.6.6 Man-in-the-Middle
671 672 673 674	Assertions with a holder-of-key subject confirmation method are not vulnerable to a MITM attack. Assertions with a sender-vouches subject confirmation method are vulnerable to MITM attacks to the degree that the receiver does not have a trusted binding of key to the vouching sender's identity.

4 Acknowledgements

- This specification was developed as a result of joint work of many individuals from
- the WSS TC including:
- 678 TBD

675

5 References 679 680 [DIGSIG] Informational RFC 2828, "Internet Security Glossary," May 2000. 681 682 [KEYWORDS] S. Bradner, "Key words for use in RFCs to Indicate Requirement 683 Levels," RFC 2119, Harvard University, March 1997 684 [SAMLBind] Oasis Committee Specification 01, P. Mishra (Editor) Bindings 685 and Profiles for the OASIS Security Assertion Markup Language 686 (SAML), May 2002. 687 [SAMLCore] Oasis Committee Specification 01, P. Hallem-Baker, and E. 688 Maler, (Editors), Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML), May 2002. 689 690 [SAMLReqs] OASIS Committee Consensus Draft, D. Platt, Evan Prodromou 691 (Editors), SAML Requirements and Use Cases, OASIS, 692 December 2001. 693 [SAMLSecure] OASIS Committee Specification 01, C. McLaren (Editor), Security and Privacy Considerations for the OASIS Security 694 695 Assertion Markup Language (SAML), May 2002. 696 [SOAP] W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 697 2000. W3C Working Draft, Nilo Mitra (Editor), SOAP Version 1.2 Part 698 0: Primer, June 2002. 699 700 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah 701 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen 702 (Editors), SOAP Version 1.2 Part 1: Messaging Framework, June 703 2002. 704 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen 705 706 (Editors), SOAP Version 1.2 Part 2: Adjuncts, June 2002. 707 [URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource 708 Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998. 709 710 Contribution to the WSS TC, P. Mishra (Editor), WS-Security [WS-SAML] 711 Profile of the Security Assertion Markup Language (SAML) 712 Working Draft 04, Sept 2002. 713 [WS-Security] TBS – point to the OASIS core draft 714 [XML-ns] W3C Recommendation, "Namespaces in XML," 14 January 715 1999.

716 717	[XML Signature	W3C Recommendation, "XML Signature Syntax and Processing," 12 February 2002.
718 719	[XML Token]	Contribution to the WSS TC, Chris Kaler (Editor), WS-Security Profile for XML-based Tokens, August 2002.
720		

721 Appendix A: Revision History

Rev	Date	What
01	19-Sep-02	Initial draft produced by extracting SAML related content from [XML token]
02	23-Sep-02	Merged in content from SS TC submission
03	18-Nov-02	Resolved issues raised by TC
04	09-Dec-02	Refined confirmation mechanisms, and added signing example
05	15-Dec-02	Results of Baltimore F2F
06	21-Feb-03	Changed name to profile

722

Appendix B: Notices

- 724 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
- 726 technology described in this document or the extent to which any license under such
- 727 rights might or might not be available; neither does it represent that it has made any
- 728 effort to identify any such rights. Information on OASIS's procedures with respect to
- 729 rights in OASIS specifications can be found at the OASIS website. Copies of claims of
- 730 rights made available for publication and any assurances of licenses to be made
- available, or the result of an attempt made to obtain a general license or permission
- for the use of such proprietary rights by implementors or users of this specification,
- 733 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
- 736 be required to implement this specification. Please address the information to the
- 737 OASIS Executive Director.

723

- 738 Copyright © OASIS Open 2002. All Rights Reserved.
- 739 This document and translations of it may be copied and furnished to others, and
- 740 derivative works that comment on or otherwise explain it or assist in its
- 741 implementation may be prepared, copied, published and distributed, in whole or in
- 742 part, without restriction of any kind, provided that the above copyright notice and
- 743 this paragraph are included on all such copies and derivative works. However, this
- document itself does not be modified in any way, such as by removing the copyright
- 745 notice or references to OASIS, except as needed for the purpose of developing
- 746 OASIS specifications, in which case the procedures for copyrights defined in the
- 747 OASIS Intellectual Property Rights document must be followed, or as required to
- 748 translate it into languages other than English.
- The limited permissions granted above are perpetual and will not be revoked by
- 750 OASIS or its successors or assigns.
- 751 This document and the information contained herein is provided on an "AS IS" basis
- 752 and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
- 753 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN
- 754 WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
- 755 MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.