SAML V2.0 Channel Binding Extensions Version 1.0

Working Draft 05

22 August 2011

Technical Committee:
OASIS Security Services TC

Chair(s):
Thomas Hardjono, M.I.T.
Nate Klingenstein, Internet2

Editor(s):
Scott Cantor, Internet2

Related Work:
This specification builds on the notion of channel bindings described in [RFC5056] and extends profiles defined in [SAML2Prof] and elsewhere.

Declared XML Namespace(s):
`urn:oasis:names:tc:SAML:protocol:ext:channel-binding`

Abstract:
Protocol extensions enable extension-aware SAML requesters and responders to modify protocol behavior in a generic, layered fashion. This specification defines an extension to the SAML V2.0 protocol [SAML2Core] specification that supports the use of channel bindings [RFC5056] in conjunction with SAML profiles. It also includes a new SAML profile that applies the extension to a set of profiles that fit a particular communication pattern.

Status:
This document is a Working Draft and as such as no official standing with regard to the OASIS Technical Committee Process.

Table of Contents

1 Introduction

Channel binding, as described in [RFC5056], is a way of associating the authentication of communicating peers at one layer of the network stack with a secure channel established at a lower level of the stack, such as TLS. This specification describes an extension that facilitates the addition of channel bindings to SAML protocol messages and assertions.

Protocol extensions consist of elements defined for inclusion in the `<samlp:Extensions>` element that modify the behavior of SAML requesters and responders when processing extended protocol messages. The protocol extension defined in this specification allows for the inclusion of channel binding information into SAML requests or responses.

A SAML V2.0 metadata [SAML2Meta] extension attribute is also defined to enable the signaling of channel binding support by particular endpoints.

Finally, a "meta"-profile is presented that acts as an extension for a variety of existing SAML profiles that fit an elementary request/response pattern.

1.1 Terminology and Notation

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119]. These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

The term *TLS* as used in this specification refers to either the Secure Sockets Layer (SSL) Protocol 3.0 [SSL3] or any version of the Transport Layer Security (TLS) Protocol [RFC2246][RFC4346][RFC5246]. As used in this specification, the term *TLS* specifically does **not** refer to the SSL Protocol 2.0 [SSL2].

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

Prefix	XML Namespace	Comments
`cb:`	urn:oasis:names:tc:SAML:protocol:ext:channel-binding	This is the SAML V2.0 channel binding extension namespace defined by this document and its accompanying schema.
`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].
`samlp:`	urn:oasis:names:tc:SAML:2.0:protocol	This is the SAML V2.0 protocol namespace defined in the SAML V2.0 core specification [SAML2Core].
`md:`	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace defined in the SAML V2.0 metadata specification [SAML2Meta].
`S:`	http://schemas.xmlsoap.org/soap/envelope/	This is the SOAP 1.1 envelope namespace defined in [SOAP1.1].
`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.

This specification uses the following typographical conventions in text: `<ns:Element>`, `Attribute`, **Datatype**, `OtherCode`.

This specification uses the following typographical conventions in XML listings:

```
Listings of XML schemas appear like this.
```

```
Listings of XML examples appear like this.  These listings are non-normative.
```

1.2 Normative References

[CBReg] Channel Binding Types Registry, IANA. http://www.iana.org/assignments/channel-binding-types/

[RFC2045] N. Freed et al. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. IETF RFC 2045, November 1996. http://www.ietf.org/rfc/rfc2045.txt

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

[RFC2246] T. Dierks, C. Allen. *The Transport Layer Security Protocol Version 1.0*. IETF RFC 2246, January 1999. http://www.ietf.org/rfc/rfc2246.txt

[RFC4346] T. Dierks, E. Rescorla. *The Transport Layer Security Protocol Version 1.1*. IETF RFC 4346, April 2006. http://www.ietf.org/rfc/rfc4346.txt

[RFC5056] N. Williams. *On the Use of Channel Bindings to Secure Channels*. IETF RFC 5056, November 2007. http://www.ietf.org/rfc/rfc5056.txt

[RFC5246] T. Dierks, E. Rescorla. *The Transport Layer Security Protocol Version 1.2*. IETF RFC 5246, August 2008. http://www.ietf.org/rfc/rfc5246.txt

[SAML2Bind] OASIS Standard, *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[SAML2Core] OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SAML2Errata] OASIS Approved Errata, *SAML V2.0 Errata*, October 2009. http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf

[SAML2Meta] OASIS Standard, Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[SAML2Prof] OASIS Standard, *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[Schema1] H. S. Thompson et al. XML Schema Part 1: Structures. World Wide Web Consortium Recommendation, May 2001. http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[Schema2] Paul V. Biron, Ashok Malhotra. XML Schema Part 2: Datatypes. World Wide Web Consortium Recommendation, May 2001. http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

[SOAP1.1] D. Box et al. *Simple Object Access Protocol (SOAP) 1.1*. World Wide Web Consortium Note, May 2000. http://www.w3.org/TR/SOAP

[SSL3] A. Freier, P. Karlton, P. Kocher. *The SSL Protocol Version 3.0*. Netscape Communications Corp., November 18, 1996. http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt

[XMLSig] D. Eastlake et al. *XML-Signature Syntax and Processing, Second Edition*. World Wide Web Consortium Recommendation, June 2008. http://www.w3.org/TR/xmldsig-core/

1.3 Non-Normative References

[RFC5929] J. Altman, et al. *Channel Bindings for TLS*. IETF RFC 5929, July 2010. http://www.ietf.org/rfc/rfc5929.txt

[SSL2] K. Hickman. *The SSL Protocol*. Netscape Communications Corp., February 9, 1995. http://www.mozilla.org/projects/security/pki/nss/ssl/draft02.html

[XMLEnc] D. Eastlake et al. XML Encryption Syntax and Processing. World Wide Web Consortium Recommendation, December 2002. See http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

2 SAML V2.0 Protocol Extension for Channel Bindings

2.1 Required Information

Identification: `urn:oasis:names:tc:SAML:protocol:ext:channel-binding`

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

Description: Given below.

Updates: None.

2.2 Overview

This extension defines a mechanism for the communication of channel bindings at the SAML protocol layer, along with a SAML metadata extension to assist in the deployment of extended capabilities. This extension allows arbitrarily defined channel binding data to be attached to a SAML request or response message (i.e., any protocol message derived from **samlp:RequestAbstractType** or **samlp:StatusResponseType**). The extension can also be used as a SOAP header block for use with more complex profiles.

Specific definitions of channel binding data are out of scope of this specification; the IANA registry can be found at [CBReg].

2.3 Element <cb:ChannelBindings>

The `<cb:ChannelBindings>` element contains typed, opaque channel bindings that are associated with a SAML request or response. The element includes the following attributes:

`Type` [optional]

A string that identifies the type of the enclosed channel bindings. Channel binding types are registered by IANA at [CBReg]. For some applications, the type of channel binding in use will be unknown to the layer that creates the extension, so this attribute is optional.

`S:actor` [optional]

Supports the element's use as a SOAP header block, unused otherwise.

`S:mustUnderstand` [optional]

Supports the element's use as a SOAP header block, unused otherwise.

The content of this element consists of application- and type-specific channel bindings, base64-encoded. The element MAY be empty. The actual content of the element must be specified by SAML profiles or other specifications that make use of this extension. Such specifications MUST ensure that the data is base-64 encoded, usually as a final encoding step.

The schema for the `<cb:ChannelBindings>` element, and its corresponding **cb:ChannelBindingsType** complex type, is as follows:

```
<element name="ChannelBindings" type="cb:ChannelBindingsType"/>
<complexType name="ChannelBindingsType">
  <simpleContent>
    <extension base="base64Binary">
      <attribute name="Type" type="string"/>
```

```
      <attribute ref="S:actor"/>
      <attribute ref="S:mustUnderstand"/>
    </extension>
  </simpleContent>
</complexType>
```

2.4 Processing Rules

This extension is included in a protocol message by placing it in the optional `<samlp:Extensions>` element. All extensions are explicitly deemed optional in SAML, so processing of the extension can never be assumed, absent additional out of band knowledge or subsequent signaling. The SAML V2.0 metadata extension defined in section 2.6 MAY be used to indicate the ability to process this extension at a particular endpoint.

There are no explicit processing requirements associated with this extension, as it is required that other profiles supply them. As a generic matter, when this element is non-empty, a message that contains this extension is considered bound to the specified channel if the message can be authenticated by means other than the specified channel, and if the message recipient can independently verify the channel bindings in a profile-specific manner.

As a simple example, normatively described in section 3, a signed SAML request containing TLS channel bindings [RFC5929] sent to a TLS-enabled endpoint can be bound to the TLS connection if the SAML responder can verify that its channel bindings match that found in the request. More complex scenarios are possible in profiles that involve active intermediaries between SAML entities.

This extension element MAY be empty, in which case it can be used to signal the successful processing/verification of channel bindings supplied by an associated message (typically identified using the `InResponseTo` attribute). For example, a response message could signal the successful verification of channel bindings supplied in the associated request.

2.5 Use Within <saml:Advice>

This extension MAY be used within the `<saml:Advice>` element to indicate that an assertion was issued in conjunction with the verification of channel bindings by the issuing authority. Either form (empty or non-empty) MAY be used. All advice elements have optional semantics, and MAY be ignored in establishing assertion validity, but relying parties MAY take into account the presence or absence of this extension in determing whether to accept an assertion.

The use of this extension within an assertion is essentially an optimization to permit signaling that would otherwise occur in a `<samlp:Response>` message to avoid signature duplication. It is analagous in that regard to data such as the `InResponseTo` or `Recipient` attributes found in the `<SubjectConfirmationData>` element.

2.6 Metadata Considerations

SAML metadata MAY be used to indicate support for this protocol extension at particular protocol endpoints, using the extension capabilities of the metadata schema.

Support for this extension is expressed in SAML V2.0 metadata [SAML2Meta] by adding an XML attribute to an element derived from the **md:EndpointType** complex type, indicating that SAML protocol messages sent to that endpoint MAY include this extension, and identifying which types of channel bindings are supported in a whitespace-delineated list.

The following schema fragment defines the `cb:supportsChannelBindings` attribute:

```
<attribute name="supportsChannelBindings">
  <simpleType>
```

```
    <list itemType="string"/>
  </simpleType>
</attribute>
```

2.6.1 Metadata Example

The example below shows a fragment of an <md:AttributeService> element that advertises support for this extension. The namespace declaration must be in scope, but the prefix is of course arbitrary.

```
<md:AttributeService
  xmlns:cb="urn:oasis:names:tc:SAML:ext:channel-binding"
  cb:supportsChannelBindings="tls-server-end-point" .../>
```

3 Use of Protocol Extension with Two-Party Profiles

3.1 Required Information

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:two-party`

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

Description: Given below.

Updates: SAML profiles designed around a simple request/response exchange between two parties.

3.2 Profile Overview

A number of SAML profiles exist that define the use of SAML request/response message pairs between a pair of entities communicating directly with each other in a simple manner. Generally such profiles are used with the SAML SOAP Binding [SAML2Bind], though this is not assumed or required. Examples of such profiles include, but are not limited to, the Artifact Resolution, Assertion Query/Request, Name Identifier Mapping, and Single Logout Profiles [SAML2Prof] (the latter in its "back-channel" form).

This profile defines an enhanced variant of all such profiles that relies on the protocol extension defined in section 2 to provide additonal security options for SAML entities supporting such profiles by binding the SAML exchange to an secure channel that is established between the parties, but not used for mutual authentication of the SAML exchange.

This is accomplished via the SAML requester attaching channel bindings to its SAML request message. The SAML responder can optionally verify the channel bindings, and adjust its behavior according to local policy (suggested examples are given below). A SAML requester could also adjust its behavior in subsequent communication with the SAML responder over the same channel.

3.3 Profile Description

3.3.1 SAML Request issued by Requesting Entity

A SAML request message is formulated and tramsitted in accordance with existing SAML profile and binding requirements, but in the presence of a secure channel for transport of the SAML binding such as TLS, the SAML requester MAY attach one or more channel bindings by including one or more `<cb:ChannelBindings>` extension elements in the SAML request's `<samlp:Extensions>` element.

Within each extension element, the `Type` attribute MUST be set to the channel binding type, and the raw channel binding data MUST be base64-encoded and the result used as the content of the element.

The SAML request MUST be integrity protected and authenticated (obviously by means other than the secure channel), typically via an XML Signature [XMLSig].

3.3.2 Verification of Channel Bindings by Responding Entity

The SAML responder SHOULD examine the `<cb:ChannelBindings>` extension element(s), if present in the SAML request, and verify at least one of the channel bindings. In the event of verification failure, the SAML responder MAY return an error/failure response to the requester. It MAY include a second-level status code of:

```
urn:oasis:names:tc:SAML:ext:channel-binding
```

If it chooses not to return an error and proceed, the SAML responder SHOULD take into account the presence or absence of channel bindings in formulating its response. In their absence, the responder MUST NOT assume a secure channel between itself and the requester. A typical example might include choosing between XML Encryption [XMLEnc] and relying on the secure channel for confidentiality.

3.3.3 SAML Response issued by Responding Entity

A SAML response message is formulated and transmitted in accordance with existing SAML profile and binding requirements. If the responder successfully verified channel bindings supplied by the requester, it MUST include at least one `<cb:ChannelBindings>` extension element in the SAML response's `<samlp:Extensions>` element, and/or in an enclosed `<saml:Assertion>`'s `<saml:Advice>` element.

The extension element(s) MAY be empty, but MUST contain a `Type` attribute indicating the type of channel bindings verified. More than one element MAY be included if the responder verified more than one type of channel bindings.

Upon receipt of the response, the SAML requester MAY apply local policy based on the presence or absence of the indication of successful verification of the channel bindings, such as adjusting its own reliance on the channel in subsequent communication.

3.4 Use of Metadata

While use of this extended variant is backwardly compatible with profile endpoints that lack such support, the metadata extension defined in section 2.6 SHOULD be used by SAML responders to indicate support for the extension, and SAML requesters SHOULD make use of the metadata extension content in deciding what type of channel bindings to supply.

3.5 Security Considerations

SAML requesters that attach channel bindings MUST ensure that the responder includes an appropriate indication of successful verification before assuming the presence of a secure channel. Since SAML is not defined in terms of connection-oriented communication, there is no preparatory "establishment" of a security context that would signal the success or failure of the channel binding separately from the SAML communication itself.

Channel bindings MAY be sent without confidentiality protection and knowledge of them is assumed to provide no advantage to an MITM.

The general security considerations of channel bindings [RFC5056] and specific channel binding types [CBReg] also apply.

4 Conformance

4.1 SAML V2.0 Protocol Extension for Channel Bindings

There are no explicit conformance requirements associated with this section, but any SAML implementation conformant with [SAML2Core] is expected to successfully process SAML messages are assertions that contain the extension (as all such extensions are explicitly optional).

4.2 Use of Protocol Extension with Two-Party Profiles

A SAML requester that supports one or more profiles compatible with the variant described in section 3.2 supports the variant/extended version of those same profiles if it conforms to the normative requirements for SAML requesters throughout section 3.

A SAML responder that supports one or more profiles compatible with the variant described in section 3.2 supports the variant/extended version of those same profiles if it conforms to the normative requirements for SAML responders throughout section 3.

Appendix A. Acknowledgments

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

- TBD

The editor would also like to acknowledge the following contributors:

- Nicolas Williams, Oracle Corporation
- Simon Josefsson, SJD AB

Appendix B. Revision History

- Working Draft 01 – Initial draft.
- Working Draft 02 – Apply new OASIS template and change filenames.
- Working Draft 03 – Fixes to template and corrected Nate's name.
- Working Draft 04 – Clarify that encoding of CB data is left to profiles, and nail down encoding for the inline profile.