

Oasis Security Services Bindings Model

Prateek Mishra
Jeff Hodges
draft-sstc-bindings-model-00.html
4-Feb-2001
comments to: security-bindings@lists.oasis-open.org

This document is an OASIS-Draft and is [largely] in conformance with relevant OASIS SSTC document standards as described in draft-sstc-doc-guidelines-00.txt.

Introduction

The purpose of this document is to (1) characterize the scope of work and deliverables for the bindings sub-committee, (2) identify relevant work items and open issues, (3) point to relevant references. It should provide a reasonably complete starting point for the efforts of the binding sub-committee.

Definitions/terminology

[JeffH: below list is just an example of the terms I've been extracting from various docs to stuff into a glossary. this isn't a definitive list. This list is interesting, though, in that they are ones that arise in the context of thinking about bindings.]

assertion (aka "security assertion"?)

authn - authentication

authz - authorization

business payload

message payload

originating site

package -- == assertions [+ entitlements] + payload ?

payload

principal

receiving site

Relying party

root -- "root of the message" (from mime?)

scrutinize

security package - one or more s2ml documents combined into a single MIME entity.

security services

subject

web service

Scope

Other Oasis Security Services TC subcommittees (e.g. Core Assertions and Protocol) are producing a specification of security assertions and services.

The high-level goal of the Bindings subcommittee is to specify how..

(1) security assertions are embedded in or combined with other objects (e.g. files of various types), communicated from site to site over various protocols, and subsequently scrutinized, and,

(2) security services are created by combining security assertions with standard messaging protocols.

(1) and (2) MUST be specified in sufficient detail to yield interoperability when independently implemented.

Deliverables

- General guidelines for *binding* security assertions to payloads in the context of a protocol.
[JeffH:I'm wondering just how distinct this is from the third item below. Perhaps the intent of this item is more: embedding security assertions into other objects (independent of protocols)? cf. S2ML 4.4]
 These should include considerations of the case where the assertions are "secret" versus the case when they are "scoped". cf. [S2ML]
- A framework for describing and registering proposed and future bindings [Open Issue-1: Need language and references for this. Preliminary text lifted from other existing protocol specifications is included below]
- Bindings for selected protocols.

Some bindings will be specified in enough detail to satisfy the interoperability requirement, while some will be merely examples of how such a binding might be done (e.g. with SyncML, FpML, cXML). The intent of the latter is that such bindings are "recommendations" ~~from~~ of the Oasis SSTC; the groups responsible for developing those protocols will be responsible for defining normative bindings with ???ML security assertions. This is facilitated by providing a method for describing and registering bindings.

- Formulating Authn and Authz *security services* by combining web service protocols such as SOAP/XP, UDDI, BEEP, and others, with security assertions. These bindings are specific instances of use of the preceding item.

Assertion Bindings

Assertion bindings will be provided for the following standard protocols:

(a) HTTP

In case of HTTP, there is a sub-case where a user is utilizing a standard off-the-shelf browser. In this case, it is important to ensure that mechanisms for conveying assertions from one site to another be developed that are based on URLs and HTTP headers (e.g., cookies). Both of these entities are strongly size constrained. Representing assertions by some form of "small" fixed-size object is an important consideration here [Section 6.1, S2ML].

[Section 6.2, S2ML] provides some discussion of a HTTP binding which is not constrained by the use of web browsers.

(b) MIME [Section 6.3 S2ML]

(c) SMTP [Open Issue-2: Relationship to (b) above] [JeffH: I seriously wonder if there are any viable use cases for a SMTP binding that aren't addressed by a definition of MIME packaging for security assertions?]

(d) ebXML

(e) SOAP/XP

(f) BEEP

Registration Templates

[JeffH: the below extracted from [BEEP] as boilerplate/example text that will need substantial massaging -- but whose underlying concepts are applicable here.]

When a profile is registered, the following information is supplied:

Profile Identification: specify a URI[10] that authoritatively identifies this profile.

Message Exchanged during Channel Creation: specify the datatypes that may be exchanged during channel creation.

Messages starting one-to-one exchanges: specify the datatypes that may be present when an exchange starts.

Messages in positive replies: specify the datatypes that may be present in a positive reply.

Messages in negative replies: specify the datatypes that may be present in a negative reply.

Messages in one-to-many exchanges: specify the datatypes that may be present in a one-to-many exchange.

Message Syntax: specify the syntax of the datatypes exchanged by the profile.

Message Semantics: specify the semantics of the datatypes exchanged by the profile.

Contact Information: specify the postal and electronic contact information for the author of the profile.

5.2 Feature Registration Template

When a feature for the channel management profile is registered, the following information is supplied:

Feature Identification: specify a string that identifies this feature. Unless the feature is registered with the IANA, the feature's identification must start with "x-".

Feature Semantics: specify the semantics of the feature.

Contact Information: specify the postal and electronic contact information for the author of the feature.

[JeffH: the below extracted from [SASL] as boilerplate/example text that will need substantial massaging -- but whose underlying concepts are applicable here.]

4. Profiling requirements

In order to use this specification, a protocol definition must supply the following information:

1. A service name, to be selected from the IANA registry of "service" elements for the GSSAPI host-based service name form [RFC 2078].
2. A definition of the command to initiate the authentication protocol exchange. This command must have as a parameter the mechanism name being selected by the client.

The command SHOULD have an optional parameter giving an initial response. This optional parameter allows the client to avoid a round trip when using a mechanism which is defined to have the client send data first. When this initial response is sent by the client and the selected mechanism is defined to have the server start with an initial challenge, the command fails. See section 5.1 of this document for further information.

3. A definition of the method by which the authentication protocol exchange is carried out, including how the challenges and responses are encoded, how the server indicates completion or failure of the exchange, how the client aborts an exchange, and how the exchange method interacts with any line length limits in the protocol.
4. Identification of the octet where any negotiated security layer starts to take effect, in both directions.
5. A specification of how the authorization identity passed from the client to the server is to be interpreted.

6. Registration procedures

Registration of a SASL mechanism is done by filling in the template in section 6.4 and sending it in to iana@isi.edu. IANA has the right to reject obviously bogus registrations, but will perform no review of claims made in the registration form.

There is no naming convention for SASL mechanisms; any name that conforms to the syntax of a SASL mechanism name can be registered.

While the registration procedures do not require it, authors of SASL mechanisms are encouraged to seek community review and comment whenever that is feasible. Authors may seek community review by posting a specification of their proposed mechanism as an internet-draft. SASL mechanisms intended for widespread use should be standardized through the normal IETF process, when appropriate.

6.1. Comments on SASL mechanism registrations

Comments on registered SASL mechanisms should first be sent to the "owner" of the mechanism. Submitters of comments may, after a reasonable attempt to contact the owner, request IANA to attach their comment to the SASL mechanism registration itself. If IANA approves of this the comment will be made accessible in conjunction with the SASL mechanism registration itself.

6.2. Location of Registered SASL Mechanism List

SASL mechanism registrations will be posted in the anonymous FTP directory "ftp://ftp.isi.edu/in-notes/iana/assignments/sasl-mechanisms/" and all registered SASL mechanisms will be listed in the periodically issued "Assigned Numbers" RFC [currently STD 2, RFC 1700]. The SASL mechanism description and other supporting material may also be published as an Informational RFC by sending it to "rfc-editor@isi.edu" (please follow the instructions to RFC authors [RFC 2223]).

6.3. Change Control

Once a SASL mechanism registration has been published by IANA, the author may request a change to its definition. The change request follows the same procedure as the registration request.

The owner of a SASL mechanism may pass responsibility for the SASL mechanism to another person or agency by informing IANA; this can be done without discussion or review.

The IESG may reassign responsibility for a SASL mechanism. The most common case of this will be to enable changes to be made to mechanisms where the author of the registration has died, moved out of contact or is otherwise unable to make changes that are important to the community.

SASL mechanism registrations may not be deleted; mechanisms which are no longer believed appropriate for use can be declared OBSOLETE by a change to their "intended use" field; such SASL mechanisms will be clearly marked in the lists published by IANA.

The IESG is considered to be the owner of all SASL mechanisms which are on the IETF standards track.

6.4. Registration Template

To: iana@iana.org
Subject: Registration of SASL mechanism X

SASL mechanism name:

Security considerations:

Published specification (optional, recommended):

Person & email address to contact for further information:

Intended usage:

(One of COMMON, LIMITED USE or OBSOLETE)

Author/Change controller:

(Any other information that the author deems interesting may be added below this line.)

Security Assertion-based Authn & Authz Services

[Section 7, Auth-XML] gives some examples of creating a security service by binding security assertions with e.g. HTTP or SOAP/XP request-response protocols.

References

[AuthXML] AuthXML: A Specification for Authentication Information in XML, Version 0.3, 12/14/2000

[BEEP] The Blocks Extensible Exchange Protocol Core <http://www.normos.org/ietf/draft/draft-ietf-beep-framework-11.txt>

[S2ML] S2ML: Security Services Markup Language, Version 0.8a, January 8, 2001.

[SASL] Simple Authentication and Security Layer (SASL) <http://www.ietf.org/rfc/rfc2222.txt>

[Shib] Shibolet Overview and Requirements

<http://middleware.internet2.edu/shibboleth/docs/draft-internet2-shibboleth-requirements-00.html>