XACML Profile for SAML

Working Draft 01, 25 February 2004

Document identifier:

wd-xacml-rbac-profile-01

Location:

http://www.oasis-open.org/committees/xacml/

Editor:

Anne Anderson, Sun Microsystems (anne.anderson@sun.com)
Hal Lockhart, BEA (hlockhar@bea.com)

Abstract:

This specification defines a profile for the use of the OASIS Security Assertion Markup Language (SAML) to carry XACML policies, policy queries and responses, authorization decisions, and authorization decision queries and responses.

Status:

This version of the specification is a working draft within the OASIS XACML TC. As such, it is expected to change prior to adoption as an OASIS standard.

Committee members should send comments on this specification to the xacml@lists.oasis-open.org list. Others should subscribe to and send comments to the xacml-comment@lists.oasis-open.org list. To subscribe, send an email message to xacml-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the XACML TC web page (http://www.oasis-open.org/committees/xacml/).

For any errata page for this specification, please refer to the XACML SAML Profile section of the XACML TC web page (http://www.oasis-open.org/committees/xacml/).

Table of Contents

1 Introduction (non-normative)

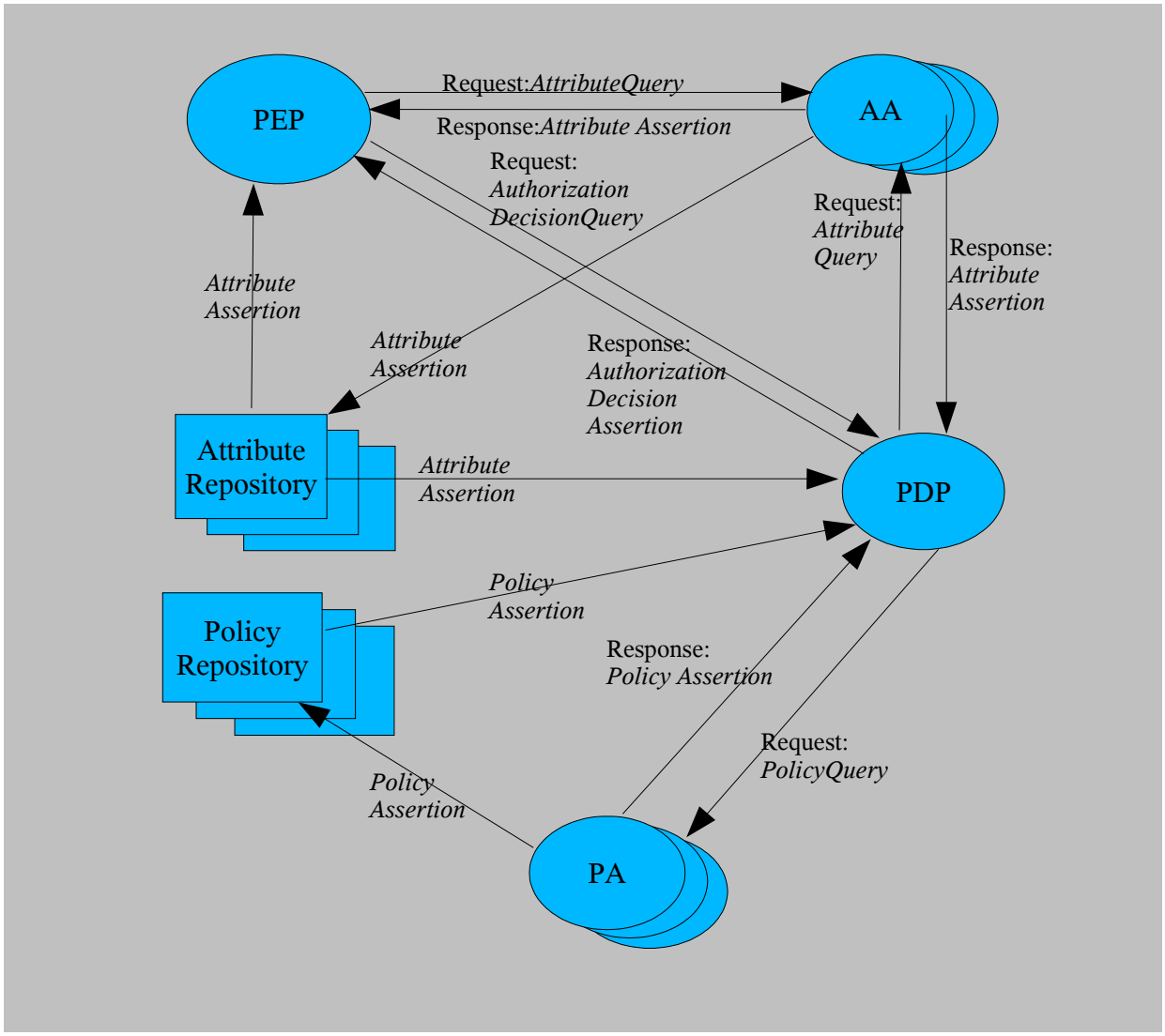
The OASIS eXtensible Access Control Markup Language [] is a powerful, standard language that specifies authorization requests, responses, policies, and policy evaluation semantics. A brief overview of XACML is available in [].

The XACML usage model assumes that a *Policy Enforcement Point* (PEP) is responsible for protecting access to one or more resources. When a resource access is attempted, the PEP sends a description of the attempted access to a *Policy Decision Point* (PDP). The PDP evaluates this request against its available policies and attributes and produces an authorization decision that is returned to the PEP. The PEP is responsible for enforcing the decision.

In producing its description of the access request, the PEP may obtain attributes from on-line *Attribute Authorities* (AA) or from *Attribute Repositories* into which AA's have stored attributes. The PDP may augment the PEP's description of the access request with additional attributes obtained from Attribute Authorities or Attribute Repositories.

The PDP may obtain its policies from on-line *Policy Authorities* (PA) or *Policy Repositories* into which PA's have stored policies.

The following diagram illustrates these interactions.



XACML itself defines some of the components necessary to implement this model, but deliberately confines its scope to the language elements used directly by the PDP and does not define protocols or transport mechanisms. Full implementation of the usage model depends on use of other standards for assertions, protocols, and transport mechanisms. XACML also does not specify how to implement a Policy Enforcement Point, a Policy Authority, or an Attribute Authority, but developers can use XACML along with other standards to provide interoperable implementations of these entities.

One standard suitable for providing the assertion and protocol mechanisms needed by XACML is the OASIS Security Markup Assertion Language []. SAML defines Query and Assertion schemas for various types of information, as well as Request and Response schemas for elements that can carry instances of a SAML Query or Assertion. The SAML schemas include information needed to identify and protect the various types of Query and Assertion payloads. SAML also has associated specifications that define bindings to other standards for providing transport mechanisms and digital signatures needed for implementation of the XACML usage model.

This specification describes extensions to SAML needed to support XACML, and also describes other aspects of using SAML with XACML. This specification requires no changes or extensions to XACML, but does define extensions to SAML.

1.1 Notation

In order to improve readability, the examples in this profile assume use of the following XML Internal Entity declarations:

```
^lt;!ENTITY xacml "urn:oasis:names:tc:xacml:1.0:">
^lt;!ENTITY xml "http://www.w3.org/2001/XMLSchema#">
^lt;!ENTITY rule-combine
             "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:">
^lt;!ENTITY policy-combine
             "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:">
^lt;!ENTITY function "urn:oasis:names:tc:xacml:1.0:function:">
^lt;!ENTITY subject-category
             "urn:oasis:names:tc:xacml:1.0:subject-category:">
^lt;!ENTITY subject "urn:oasis:names:tc:xacml:1.0:subject:">
^lt;!ENTITY resource "urn:oasis:names:tc:xacml:1.0:resource:">
^lt;!ENTITY action "urn:oasis:names:tc:xacml:1.0:action:">
^lt;!ENTITY environment "urn:oasis:names:tc:xacml:1.0:environment:">
```

For example, `&xml;#string` is equivalent to `http://www.w3.org/2001/XMLSchema#string`.

1.2 Terminology

The key words *must, must not, required, shall, shall not, should, should not, recommended, may,* and *optional* in this document are to be interpreted as described in IETF RFC 2119 [RFC2119].

attribute - In this Profile, the term “attribute” refers to an XACML `<Attribute>`. An XACML `<Attribute>` is an element in an XACML Request having among its components an attribute name identifier, a data type identifier, and an attribute value. Each `<Attribute>` is associated either with one of the subjects (Subject Attribute), the protected resource (Resource Attribute), the action to be taken on the resource (Action Attribute), or the environment of the Request (Environment Attribute). Attributes are referenced in a policy by using an `<AttributeSelector>` (an XPath expression) or one of the following: `<SubjectAttributeDesignator>`, `<ResourceAttributeDesignator>`, `<ActionAttributeDesignator>`, or `<EnvironmentAttributeDesignator>`.

PDP - Policy Decision Point. An entity that evaluates an access request against one or more policies to produce an access decision.

PEP – Policy Enforcement Point. An entity that enforces access control for one or more resources. When a resource access is attempted, a PEP sends an access request describing the attempted access to a PDP. The PDP returns an access decision that the PEP then enforces.

policy – A set of rules indicating which subjects are permitted to access which resources using which actions under which conditions.

2 Attributes (normative)

For attributes, SAML defines AttributeQuery and Attribute Assertion schemas. A SAML Request and Response can carry instances of these as part of the SAML protocol payload. XACML implementations can use instances of these SAML schemas to request, transmit, and store Attributes. The Attribute schema definitions used by SAML and by XACML differ somewhat, however, so a mapping between them needs to be defined. This Section describes that mapping.

2.1 Mapping a SAML Attribute Assertion to XACML Attributes

[Show the two schemas and describe how the elements map. One specific item is that the Issuer of the SAML Attribute Assertion is the Issuer of each XACML Attribute derived from that Attribute Assertion.]

2.2 XSLT for the mapping

[Michiharu produced a mapping that might be usable here.]

3 Authorization Decisions (normative)

For authorization decisions, SAML defines very rudimentary AuthorizationDecisionQuery and AuthorizationDecision Assertion schemas. XACML defines RequestContext and ResponseContext schemas that describe an authorization decision request and response, respectively. The SAML and XACML schemas differ, but it is possible to use XSLT [] to map a SAML AuthorizationDecisionQuery to an XACML RequestContext and to populate a SAML AuthorizationDecision from an XACML ResponseContext. This Section describes that process.

A SAML AuthorizationDecisionQuery is unable to convey all the information that an XACML PDP is capable of acceptiing as part of its RequestContext. Likewise, the SAML AuthorizationDecision Assertion is unable to convey all the information contained in an XACML ResponseContext. To solve this problem, this specification defines new SAML extensions for XACMLAuthorizationDecisionQuery and XACMLAuthorizationDecision Assertion schemas that allow a PEP to use the full capabilities of an XACML PDP. This Section also defines these extensions.

3.1 XACMLAuthorizationDecisionQuery

[XACML 2.0 Work Item #47] Agreement between SSTC and XACML recorded in http://lists.oasis-open.org/archives/xacml/200309/msg00039.html. Since then it was decided that the extension would be defined by the XACML TC as a specific XACMLAuthorizationDecisionQuery extension to SAML, rather than as a new SAML AuthorizationDecisionQuery format.

3.2 XACMLAuthorizationDecision Assertion

[XACML 2.0 Work Item #47] See above.

4 Policies (normative)

For policies, XACML defines two policy schema elements: Policy and PolicySet, but SAML does not define any Query or Assertion schemas for policies. This Section defines new SAML extensions for PolicyQuery and Policy Assertion schemas. Instances of these new extensions can be used to request, transmit, and store XACML Policy and PolicySet instances.

4.1 XACMLPolicyQuery

[This is XACML 2.0 Work Item #40] Need to be able to query for specific PolicyId or PolicySetId. Also need to be able to supply a Request Context and get back applicable policies.

Issue: define abstract SAML PolicyQuery as part of SAML 2.0, of which XACMLPolicyQuery is an extension, or just specific XACMLPolicyQuery? In either case, XACMLPolicyQuery will be defined as a SAML extension by the XACML TC in the XACML namespace.

4.2 XACMLPolicy Assertion

[This is XACML 2.0 Work Item #40] Need to define a SAML Assertion whose payload is an XACML Policy or PolicySet. May want to include other information from the PolicyQuery.

Issue: define abstract SAML Policy Assertion as part of SAML 2.0, of which XACMLPolicyAssertion is an extension, or just specific XACMLPolicyAssertion. In either case, XACMLPolicyAssertion will be defined as a SAML extension by the XACML TC in the XACML namespace.

5 References

5.1 Normative References

[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

[XACML] T. Moses, ed., *OASIS eXtensible Access Control Markup Language (XACML) Version 1.1*,http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf, Committee Specification, 24 July 2003.

5.2 Non-normative References

[XACMLIntro] A Brief Introduction to XACML, http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html, 14 March 2003.

A. Acknowledgments

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

- *Frank Siebenlist, Argonne National Laboratory*
- *Daniel Engovatov, BEA Systems, Inc.*
- *Hal Lockhart, BEA Systems, Inc.*
- *Tim Moses, Entrust*
- *Maryann Hondo, IBM*
- *Michiharu Kudo, IBM*
- *Michael McIntosh, IBM*
- *Anthony Nadalin, IBM*
- *Rebekah Lepro, NASA*
- *Steve Anderson, OpenNetwork*
- *Simon Godik, Overxeer*
- *Bill Parducci, Overxeer*
- *Anne Anderson, Sun Microsystems*
- *Seth Proctor, Sun Microsystems*
- *Polar Humenn, Syracuse University*

In addition, the following people made contributions to this specification:

- *Ravi Sandhu, George Mason Univ.*
- *John Barkley, NIST*
- *Ramaswamy Chandramouli, NIST*
- *David Ferraiolo, NIST*
- *Rick Kuhn, NIST*
- *Serban Gavrila, VDG Inc.*

B. Revision History

Rev	Date	By Whom	What
01	12 Feb 2004	Anne Anderson	Document in Committee Draft format created from the Working Draft at http://www.oasis-open.org/committees/download.php/2405/wd-xacml-rbac-profile-01.doc

C. Notices

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 technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of 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, 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 be required to implement this specification. Please address the information to the OASIS Executive Director.