Use of SAML for OGSA Authorization

Status of This Memo

This document has been submitted to the Global Grid Forum OGSA Security Working Group for consideration as recommendations document in that area of OGSA authorization.

The latest version of this document can be found at:

http://www.globus.org/ogsa/Security/

Copyright Notice

Abstract

This document defines an open grid services architecture (OGSA) authorization service based on the use of the security assertion markup language (SAML) as a format for requesting and expressing authorization assertions. Defining standard formats for these messages allows for pluggability of different authorization systems using SAML.

Open Issues in this Document

1. **Intro: Make sure syncs with overview document (Owner: VW)**
2. **Section 3: Reorganize into Query and Response (Owner: VW)**
3. **Section: 6.1.3.2 SDE expression isn't right. Need hierarchical resource specification? (Owner: VW)**
4. **Section 6.1.4: Need to define Evidence element schemas for credentials (Owner: VW)**
5. **Section 6.1.4: Need explanation of including credentials (Owner: VW)**
6. **Section 6.1.4: Need explanation of Date,Time,DateTime Evidence. Why are they there, what should PDP do with them (Owner: DC)**
7. **Section 6.2.1: Can we say anything better about Conditions? (Owner: VW)**
8. **Section 6.2.2: Need to verify section is correct and remove comment questioning such (Owner: DC)**
9. **Section 6.2.3: Indeterminate requires Condition? VW says No. (Owner: DC)**
10. **Section 6.2.6 (and 6.1.4): Resolve differences between xsd:dateTime and ISO 8601 (Owner: DC)**
11. **Section 7: Finish flushing out GWSDL (Owner: VW)**
12. **Glossary: Make sure terms sink with overview and authz framework (Owner: VW)**
13. **References: Complete and verify (Owner: VW)**

Table of Contents

1. Introduction

There are a number of authorization systems currently available for use on the Grid as well as in other areas of computing, such as Akenti [Akenti], CAS [CAS], PERMIS [PERMIS], VOMS [VOMS] and Cardea [Ref needed]. Some of these systems are normally used in *decision push mode* by the application [RFC2904] - they act as services and issue their authorization decisions in the form of authorization assertions that are conveyed, or pushed, to the target resource by the initiator. Others are used in *decision pull mode* by the application - they are normally linked with an application or service and act as a policy decision maker for that application, which pulls a decision from them.

On the abstract level both of these types of authorization services have similar semantics - they are given a description of the initiator (which might include the initiator's privileges), a description of an action being requested (including its argument), details about the target resource to be accessed, and any contextual information such as time of day, and they provide an authorization decision whether the action should be processed or rejected.

These authorization services can themselves act in *credential push or pull mode* [RFC3281]. In *credential push mode*, the client provides all the information necessary for a decision to be made. In *credential pull mode*, the client provides everything except the initiator's privileges, and the authorization service then pulls these privilege tokens (or credentials) from some other authority, and bases its decision on them. The client may provide a pointer to the authorization service, giving it a hint where to find the privileges, or the authorization service may be pre-configured with knowledge about where to locate them.

With the emergences of OGSA and Grid Services, it is expected that some of these systems will become OGSA authorization services as mentioned in the OGSA Security Roadmap [Roadmap]. OGSA authorization services are Grid Services providing authorization functionality over an exposed Grid Service portType. A client sends a request for an authorization decision to the authorization service and in return receives an authorization assertion or a decision. A client may be the resource itself, an agent of the resource, or an initiator or a proxy for an initiator who passes the assertion on to the resource.

This specification defines the use of SAML [SAML] as a message format for requesting and expressing authorization assertions and decisions from an OGSA authorization service. The SAML AuthorizationDecisionQuery element is defined as the message to request an authorization assertion or decision, the DecisionStatement element is defined as the message to return a simple decision, and the AuthorizationDecisionStatement the method for expressing an authorization assertion. By defining standard message formats the goal is to allow these different authorization services to be pluggable to allow different authorization systems to be used interchangeably in OGSA services and clients.

Section 2 describes the conventions and namespaces used in this document. Section 3 contains a non-normative overview of the authorization portions of the SAML specification. Section 4 contains a non-normative description of SAML extensions defined in this document and Section 5 is a normative definition of those extensions. Section 6 is normative and defines how SAML elements should be used to form OGSA authorization assertions and requests. Section 7 contains the WSDL for the authorization service portType. Section 1 contains non-normative commentary. The specification concludes with GGF copyright and intellectual property statements, author affiliation and contact information and a glossary.

2. Conventions use in this Specification

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 RFC-2119 [RFC2119].

This specification uses namespace prefixes throughout; they are listed in Table 1. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 1: Namspaces used in this specification.

Prefix	Namespace
ogsa-saml	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/
operation	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/operation
sde-read	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/read
sde-modify	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/modify
wildcard	http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/wildcard
saml	urn:oasis:names:tc:SAML:1.0:assertion
samlp	urn:oasis:names:tc:SAML:1.0:protocol

3. SAML Authorization Overview

The SAML specification [SAML] defines a number of elements for making assertions and queries regarding authentication, authorization decisions and attributes. It also supports extensibility by allowing applications to define their own elements. In this section we give a brief **non-normative overview** of the elements related to authorization, and the additional elements needed for Grid authorization. Readers are encouraged to review the SAML specification for more details.

3.1 SAML Version

This specification is based on the SAML v1.1 specification. This specification defines a number of extensions to SAMLv1.1 which are described in Section 4, that are necessary for Grid authorization,. The authors are aware that OASIS is currently working on SAMLv2.0. Indeed, the authors are working closely with the OASIS organization to help ensure that SAMLv2.0 contains the extensions described in this specification (and if not syntactically identical, then at least having the same semantic content). Once SAMLv2.0 has been published, it is the intention of the authors to migrate this specification to SAMLv2.0.

3.2 SAML Authorization Model

As shown in Figure 1, SAML defines a message exchange between a policy enforcement point (PEP) and a policy decision point (PDP) consisting of an AuthorizationDecisionQuery (2) flowing from the PEP to the PDP, with an Assertion returned containing some number of AuthorizationDecisionStatements (3). We also define an extension to SAML to support exchanges in which a client can issue an AuthorizationDecisionQuery to a server, and have an Assertion returned containing a simple AuthorizationDecision.

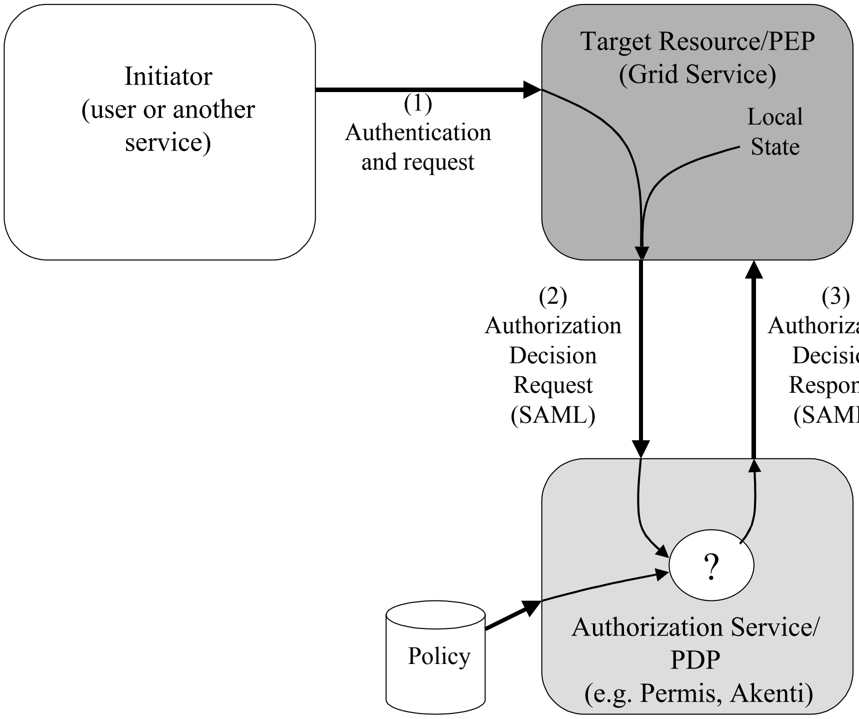
Initiator
(user or another service)
(1) Authentication and request
Target Resource/PEP
(Grid Service)
Local State
(2) Authorization Decision Request (SAML)
(3) Authorization Decision Response (SAML)
?
Policy
Authorization Service/ PDP
(e.g. Permis, Akenti)

Figure 1: SAML message flow. (1) A request arrives at the target resource. (2) The Grid Service generates and sends a SAML AuthorizationDecisionQuery to an Authorization Service. (3) The service evaluates the request against policy and returns a response encoded as a SAML Assertion.

In the following sections we describe the AuthorizationDecisionQuery and the Assertion element, and the elements that are used to compose these.

3.3 Action Element

The Action elements allows for the expression of actions that may be attempted by entities and expressed in policy. This element consists of a string and a URI defining a namespace for the action described in the string.

For example the SAML specification defines a namespace for HTTP operations that defines actions of GET, HEAD, PUT, POST.

3.4 Resource Element

The Resource element is used to identify the target on which the policy is being asserted or requested. This element is simply a URI.

3.5 Subject and NameIdentifier Elements

The Subject element contains a NameIdentifier element as well as some elements outside the scope of this document. In SAML authorization assertions, the NameIdentifer element serves to identify the initiator of the action being authorized. The NameIdentifer element contains a string to hold an identity that has two attributes:

- The NameQualifier attribute is a string expressing the security or administrative domain that defined the name (e.g. Kerberos realm, CA name).

- The Format attribute is a URI identifying the format of the name (e.g. X509 subject name).

3.6 AuthorizationDecisionStatement Element

The AuthorizationDecisionStatement element contains statements regarding authorization policy. Each of these statements contains a *Subject* element, identifying the entity whose rights are being expressed, a *Resource* element, identifying the resource(s) the rights apply to, an optional *Evidence* element holding the assertions the issuer relied upon in making its decision, any number of *Action* elements (expressing the allowed or denied operations) and the *Decision* attribute containing the authorization decision. The assertion may also optionally contain a *Conditions* element expressing the conditions that must be fulfilled before the authorization can be permitted and an *Advice* element providing additional information related to the authorization decision which may be ignored by the recipient.

3.7 AttributeStatement Element

This element supplies a statement by the issuer that the specified subject is associated with the specified attribute(s).

3.8 Assertion Element

The Assertion element specifies the basic information that is common to all SAML assertions, and optionally it may be signed. It can contain any number of Statements, for example, AuthorizationDecisionsStatements and AttributeStatements. It is also capable of containing statements related to authentication, but for the purposes of this document we only consider Assertions containing AttributeStatements, AuthorizationDecisions and AuthorizationDecisionStatements.

3.9 Conditions Elements

Each Assertion element may contain any number of Conditions elements. Conditions elements are specified to express policy restrictions on the assertion such as a validity time of the Assertion. However they are extendable to express arbitrary conditions on the use of the assertion. Condition elements might typically be added to assertions if the decision engine had insufficient information to be able to evaluate the policy locally.

3.10 Advice Elements

An Assertion element may contain any number of Advice elements. Advice elements hold information related to the assertion, but they may be ignored by applications that do not support them. Examples of information that could be included in an Advice element are: an identifier of the policy that was used by the PDP when making its authorization decision, and assertions that were used by the PDP when making its authorization decision.

3.11 AuthorizationDecisionQuery Element

The AuthorizationDecisionQuery element allows for the request of AuthorizationDecisionStatements and simple AuthorizationDecision responses. It contains a Subject, Resource, optional Evidence, and any number of Action elements that identify the decisions that the initiator wants to be made; as well as a RespondWith element that identifies the type of response that the client wishes to be returned.

3.12 Evidence Elements

Evidence elements allow for queries to provide information to the PDP that may be useful for its decision-making. They are used to hold the credentials of the initiator, as well as contextual and environmental information. The initiator’s credentials may be either included directly in the evidence element (as AttributeStatements), or may be included indirectly via a pointer (as ReferenceStatements). This allows the PDP to support both the credential push and pull mode of operation. In responses, they also allow the PDP to express what information it used to make its decision.

Each AuthorizationDecisionStatement and AuthorizationDecisionQuery element can contain any number of Evidence elements. Each Evidence element can contain any number of Assertions elements (or references to Assertion elements) that affect the policy decision process.

3.13 ReferenceStatement Element

This element allows Authorization Decision Queries to contain a pointer to an external resource, which may contain credentials for the initiator. This is used to flag the *credential pull mode* of operation.

4. Overview of Extensions

This section provides **non-normative** discussion of the extensions in this specification.

The goals of these extensions are to allow an entity requesting an authorization decision to indicate the following desires in regards to the response and for the responder to oblige those requests if it can and desires:

- To request a simple decision in regards to that query instead of a list of allowed rights of the subject.
- To request either the assertion(s) or response be signed.
- To provide one or more URIs for which attributes regarding the subject may be obtained.

4.1 Simple Authorization Query Response: New Statement Type

In the SAML authorization query protocol, a resource normally sends a query to the decision service with an enumeration of the actions being attempted by a requestor. The decision service responds with an assertion containing the set of actions that the requestor is authorized to perform.

While this functions well for situations where the resource may be interested in knowing what subset of the actions the requestor is allowed to perform, in "all or nothing" situations where the resource is only interested in knowing if the requestor can perform all the enumerated actions, it requires the resource to process the entire list to verify if all the actions originally requested are listed.

This specification defines an new StatementType, the SimpleAuthorizationDecisionStatement element, which contains a reference to the original AuthorizationDecisionQuery and a simple boolean decision in regards to that query as a whole. This allows an easy-to-parse decision to be rendered on the query as a whole, as well as potentially significantly reducing the bandwidth needed to transmit the decision.

4.2 Extenteded Authorization Query

This document defines an extended authorization query type which adds the following features:

- A mechanism to allow a requestor to indicate their interest in a simple authorization response (as described in the previous section) rather than a full set of AuthorizationDecisionStatements.
- A mechanism to allow a requestor to pass information to the PDP which it may choose to use in making in decision. This document also defines once such element, which allows a requestor to pass a pointer to the source of attribute information regarding the subject.
- A mechanism to allow a requestor to indicate their preference in regards to whether the response is signed and how. This is useful for saving work on the PDP in situations where some clients may be passing the response on to another party (e.g. in a push mode of operation) while others will be direct consumers and hence don’t need any signatures when the transport layer provides sufficient security.

5. SAML Extensions

This section is **normative**. It defines the SAML extensions used by OGSA. See the previous section for a non-normative description of these extensions.

These extensions are made to the SAML 1.1 schema using the type derivation method as described in Section 6.3 of [SAML].

5.1 Element <ExtendedAuthorizationDecisionQuery>

The ExtendedAuthorizationDecisionQuery element allows the entity making the query to indicate preferences in the query reply.

An ExtendedAuthorizationDecisionQuery element contains the following attributes:

RequestSimpleDecision [Optional]

> This elements indicates that the requestor’s preference in regards to having the response in the form of a single SimpleAuthorizationDecisionStatement (as defined in this document) instead of as one or more AuthorizationDecisionStatment elements.

Recipient [Optional]

> This element is used to indicate the intented recipient of the response. When a SimpleAuthorizationDecisionStatement is requested, it will be included in that statement to help prevent replay of such an element to entity other than the intended.

RequestSigned [Optional]

> This element is used to request that a signature be included with the response. This element should contain the name of the element to be signed i.e. samlp:Response or saml:Assertion. A responder to a query with this attribute set SHOULD sign the response as request, however is under no obligation to and MAY return a unsigned response (or one signed differently if unable or unwilling to accommodate the ReqestSigned element.

An ExtendedAuthorizationDecisionQuery element contains the following elements:

AuthorizationAdvice [Optional]

> This abstract element allows for additional information to be included with the query that the responder MAY use when rendering a decision.

The following schema franment defines the <ExtendedAuthorizationDecisionQuery> element and its ExtendedAuthorizationDecisionQueryType complex type:

```
<element name="ExtendedAuthorizationDecisionQuery"
type="ExtendedAuthorizationDecisionQueryType"/>
<complexType name=" ExtendedAuthorizationDecisionQueryType">
    <complexContent>
       <extension base="samlp:AuthorizationDecisionQuery">

          <attribute name="RequestSimpleDecision" type="boolean" use="optional"
             default=”false”/>

          <attribute name=”Recipient” type=”anyURI” use=”optional”/>

          <attribute name=”RequestSigned” type=”QName” use=”optional”/>
          <sequence>

             <element ref=”ogsa-saml:AuthorizationAdvice” minOccurs=”0”/>

          </sequence>

       </extension>
    </complexContent>
</complexType>
```

5.1.1 Element < SubjectAttributeRefeenceAdvice>

The <AuthorizationAdvice> element is an extemsion point that allows for additional information to be included with an authorization query that MAY be used by the responder.

The following scheme fragment define the <AuthorizationAdvice> element and its AuthorizationAdviceAbstractType complex type:

```
<element name=”AuthorizationAdvice” type=”ogsa-saml:AuthorizationAdviceAbstractType”/>
<complexType name=”AuthorizationAdviceAbstractType” abstract=”true”/>
```

5.1.2 Element <SubjectAttributeRefeenceAdvice>

The <SubjectAttributeReferenceAdvice> element supplies a statement by the issuer that the designated attributes associated with the specified subject may be obtained from the referenced URI. Its purpose is to advise the PDP where it may find attributes associated with the subject, and it is used to support the *credential pull mode* of operation.

<SubjectAttributeRefeenceAdvice> is of type SubjectAttributeRefeenceAdvice Type, which extends the SubjectAttributeRefeenceAdvice AbstractType with the addition of the following:

AttributeDesignator [Any number]

> These elements list the attributes that may be located at the referenced URI. If this component is absent, then it implies that all attributes can be found at the referenced URI.

Reference Attribute [Required]

> This attribute provides the URI from which the attributes may be obtained.

The following schema franment defines the <SubjectAttributeReferenceAdvice> element and its AubjectAttributeReferenceAdviceType complex type:

```
<element name="SubjectAttributeReferenceAdvice"
   type="ogsa-saml: SubjectAttributeReferenceAdviceType"/>
<complexType name="SubjectAttributeReferenceAdviceType">
      <complexContent>
          <extension base="AuthorizationAdviceType">
              <sequence>
                  <element ref="saml:AttributeDesignator" minOccurs=”0” maxOccurs=”unbounded”
/>
             </sequence>
             <attribute name="Reference” type=”anyURI” use="required"/>
         </extension>
    </complexContent>
</complexType>
```

5.2 Element <SimpleAuthorizationDecisionStatement>

The <SimpleAuthorizationDecisionStatement> element specifies the decision made about a corresponding SAML AuthorisationDecisionQuery request. Its purpose is to allow a response to the statement as a whole without enumeration of the rights in the response, which in turns allows for easier processing of the response by the requestor.

It has the complex type SimpleAuthorizationDecisionStatementType, which extends the StatementAbstractType by adding the following to it:

Decision [Required]

The decision made by the responder.

InResponseTo [Required]

The RequestID from the query which this statement is in response to. This attribute MUST be present and its value MUST match the value of the RequestID field which this statement is in response to.

Recipient

If the ExtendedAuthorizationDecisionQuery that this Statement is in response to, contained a Recipient attribute, this attribute MUST be present and its value MUST match the value of the Recipient field in the query which this statement is in response to.

The following schema franment defines the <SimpleAuthorizatonDecisionStatement> element and its SimpleAuthorizationDecisionStatementType complext type:

```
<element name="SimpleAuthorizationDecisionStatement"
type="SimpleAuthorizationDecisionStatementType"/>
<complexType name="SimpleAuthorizationDecisionStatementType">
    <complexContent>
        <extension base="saml:SubjectStatementAbstractType">
            <attribute name="Decision" type="saml:DecisionType" use="required"/>

            <attribute name=”InResponseTo” type=”NCName” use=”required”/>

            <attribute name=”Recipient” type=”anyURI” use=”optional”/>
       </extension>
   </complexContent>
</complexType>
```

6. SAML Authorization Element Usage in OGSA

This section is **normative**. It describes how SAML Authorization elements are used to meet OSGA requirements for authorization assertions and decisions as described in [Authz]. It first describes the use of the AuthorizationDecisionQuery and ExtendedAuthorizationDecisionQuery elements, which is used by entities to request authorization assertions or decisions from an authorization service. This is followed by a description of the statements that can be returned in the response, either one or more standard AuthorizationDecisionStatement elements or a SimpleAuthorizationDecisionStatement element.

6.1 (Extended)AuthorizationDecisionQuery

A client MUST request an authorization decision using either a AuthorizationDecisionQuery or an ExtendedAuthorizationDecisionQuery. This section describes constraints on fields that are in both of these elements. Fields solely in an ExtendedAuthorizationDecisionQuery are described in Section 5.1.

The AuthorizationDecisionQuery element MUST include the following elements:

- A *Subject* element containing a *NameIdentifier* element specifying the identity of the initiator.
- A *Resource* element specifying the resource (or domain of resources) to which the request to be authorized is being made.
- One or more *Action* elements specifying the action(s) being requested on the resource(s).

The query MAY include the following elements:

- Optionally one or more *Evidence* elements containing one or more supporting credentials about the initiator (or pointers to them), plus any contextual information, plus a public key certificate chain that may be used to authenticate the initiator.

The following subsections describe the use of and extensions to these elements for OGSA.

6.1.1 Subject Element

This element contains the name of the initiator. The Subject and contained NameIdentifer elements are unchanged from the SAML specification. The exact use of these elements is driven by the authentication mechanism used by the client. In some scenarios, the authorization service (PDP) MAY require the initiator and client names to be the same. In other scenarios, the authorization service MAY allow trusted clients to request authorization decisions on behalf of any initiator.

6.1.1.1 Proxy Certificate Authentication Method Identifier

The SAML specification defines how some common identity types are asserted. This document defines how entities authenticated using X.509 Proxy Certificates [ProxyCerts] should be encoded. The SAML specification defines a URI for X.509 subject names (urn:oasis:names:tc:SAML:1.1nameid-format:X509SubjectName) that MUST be used for X.509 Proxy Certificate authenticated identities with the subject name of the end entity certificate that issued the proxy certificate chain as the identity.

6.1.1.2 Wildcard Subject Identifier

This document defines a method to be used in order to obtain public rights, that is, rights available to any subject. To indicate that such a request is being made, the NameIdentifier element MUST be specified as follows:

```
http://www.gridforum.org/ogsa-authz/saml/2003/06/NameIdentifier/any
```

6.1.2 Resource Element

The Resource element is defined as a URI.

6.1.2.1 Grid Services

If the resource being referred to is a Grid service the resource element MUST contain the Grid Service Handle (GSH) of the service as described in [OGSI].

It is also possible that this element could contain a URI referring to things other than GSHs in an OGSA context. For example, a URI could be used to refer to a group of services. However such usage is determined by prior agreement between authorization services, policy makers and resources in a particular domain and is beyond the scope of this document.

6.1.2.2 Wildcard Resource

This specification also defines a wildcard resource. This has two different meanings depending on whether it is in a query (request to a PDP) or a statement (response from a PDP):

- In an AuthorizationDecisionQuery, it states a desire to learn the initiator's rights on all the resource of which the authorization service is aware. Typically such a query will be used by an initiator who will cache the results and present them to resources later in a *decision push mode* of authorization.
- In an AuthorizationDecisionStatement, it states the initiator has the given privileges on all resources that accept the authorization service as authoritative. This statement may be used when the authorization service is the authority for a group of resources with identical policy.

This wildcard URI MUST be specified as follows:

```
http://www.gridforum.org/ogsa-authz/saml/2003/06/resource/any
```

6.1.3 Action Elements

The Action element describes the operation or method to be authorized. The Action element is composed of a string describing the operation and a URI specifying the namespace of the action.

6.1.3.1 Grid Service Operation Invocation

This specification defines the following namespaces:

http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/operation

This namespace is used to define an operation invocation on the specified Resource by the specified Subject. The action string should contain the namespace and name of the operation being invoked.

Comment: This seems wrong. Why specify two namespaces for an operation? Either the grid namespace above can pre-define several operations to be used as action strings e.g. “Execute”, “Print”, “Pause”, “Resume” etc, or grid applications can specify their own namespaces and action strings.

6.1.3.2 Grid Service SDE Access

I agree with David’s comment, this is screwed up. We need some way of specifying a hierarchical resource of GSH and SDE and not overloading Action - VW

http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/read

This namespace is used to define the reading of a ServiceDataElement. The action string should contain the QName of the Service Data element being accessed.

http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/sde/modify

This namespace is used to define the modification of a ServiceDataElement. The action string should contain the QName of the Service Data element being modified.

6.1.3.3 Wildcard Action

This specification also defines a wildcard action. This action has two different meanings depending on whether it is in a query or an assertion:

- In an AuthorizationDecisionQuery, it states a desire to learn all of the initiator’s rights on the specified resource. An example of where this might be used, is by a policy enforcement point co-located with a resource, that after an intiator has set up a session, will cache the results, and do further policy processing without the authorization service.

- In an AuthorizationDecisionStatement, it states the initiator has all privileges on the resource. This will often be the case where the initiator is the policy authority for the resource in question.

This wildcard action MUST be specified as follows. The namespace URI MUST be:

http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/action/wildcard

The Action sting must be "*", i.e., an asterisk.

6.1.4 Evidence Elements

Evidence elements are assertions used to hold, either directly or by reference, supporting credentials regarding the initiator, as well as environmental parameters.

An Evidence element may hold for example an Attribute Assertion that contains the role of the initiator, or the groups that the initiator is a member of.

In the *credential push mode* of operation this element SHOULD contain the credentials of the initiator. If the initiator does not have any credentials (for example, if default or public access rights are being requested) then there will be no evidence assertions in which the subject name is that of the initiator.[1]

When the credentials are in the form of attributes, the precise way in which these are inserted into the AttributeStatements embedded in the Evidence element is specified in [Attributes]

[1] Editor’s Note. Alternatively we can indicate that the initiator has no credentials, by setting this element to <AssertionIDRefence>, and the value of the string to “null”.

In the *credential pull mode* of operation the Evidence element MAY contain a Reference Statement. The precise contents of the Reference Statement are described below.

If the client wishes the PDP to operate in both credential push and pull mode, then it MAY include initiator credentials and Reference Statements in the Evidence element. If neither is present, then it is at the discretion of the PDP how to behave (e.g. it may be pre-configured with a resource from which to pull initiator credentials, or it may assume the initiator has no credentials).

When the Evidence element is used to hold environmental parameters, these MAY be encoded up as Attribute Statements as follows.

The application MAY specify its own AttributeNamespace URI, along with AttributeName strings to represent environmental parameters (e.g. “accountCode”, “callingAddress”, “currentTime”), and appropriate environmental values for each of the AttributeNames (e.g. “ABC123”, “87.80.7.56”, “12:02:35”).

The following namespace MAY be used to specify a standard set of environmental parameters:

```
http://www.gridforum.org/namespaces/2003/06/ogsa-authz/saml/env
```

The following AttributeName strings are defined, along with the syntax for their AttributeValues:

AttributeName	AttributeValue syntax	AttributeValue Example
Date	ccyy-mm-dd	2003-02-12
Time	hh:mm:ss	12:05:35
DateTime	ccyy-mm-ddThh:mm:ss[2]	2003-02-12T12:05:35
Any others???		

The Evidence element MAY also contain the X.509 public key certificate chain that was or can be used to authenticate the initiator of the authorization decision request. How this is encoded is TBD.

This specification makes no further constraints on the use of this element for specifying credentials. It is expected that specifications for different types of supporting credentials will be developed.

6.1.5 ReferenceStatement Element

Reference statements MAY be included within Evidence elements, in order to signal the *credential pull mode* of operation to the PDP. Reference statements MAY be included instead of, or as well as, credentials in Evidence elements, and it is a local matter for the PDP to determine how to handle the presence of one, both or neither elements.

If a Reference statement is present, then the Format attribute of the NameIdentifier element of the Subject element of the Reference statement SHOULD be #X509SubjectName, and the value MUST correspond to that of the Subject element of the AuthorizationDecisionQuery.

The value of the Reference URI is not further constrained by this specification.

6.2 Assertion Element

The SAML Assertion element is used by one entity to assert the capabilities of another. While an Assertion element can contain a variety of SAML statements, for the purposes of this document we consider only AuthorizationDecisionStatements, SimpleAuthorizationDecisionStatements

[2] This is ISO 8601 format

(defined in this document) and AttributeStatements. The first two may be returned in response to AuthorizationDecisionQueries, whilst the latter may be presented in the Evidence elements of (Extended)AuthorizationDecisionQueries.

When returned by an authorization service to an entity, the Assertion element will be enveloped in a SAML Response element as described in the SAML specification.

The Assertion element includes the following elements:

- An optional *Conditions* element specifying the conditions for use of the assertion.
- An optional *Advice* element specifying advice for use of the element.
- Any number of *AuthorizationDecisionsStatements*
- Any number of *AttributeStatements* in Evidence elements
- An optional *Signature* element allowing the Assertion to be verified.

The following subsections describe the use and extensions to these elements for OGSA.

6.2.1 Conditions Element

Implementations are advised to be conservative in their use of this element and only include it when they are confident it will be understood.

The Conditions element contains optional time constraints and any number of Condition elements (note difference in plurality between element names) on the returned assertion. Condition elements serve as an abstract element for extension, and should be used to express the policy conditions on operands and context/environment that the authorization service was unable to evaluate due to insufficient information being provided by the client. It is envisioned that future specification will be able to extend the Condition element to return fine-grained policies for parameters on operation invocation and service data access, using for example elements of XACML.

6.2.2 Advice Element

The Advice element MAY be ignored by the recipient of the assertion, therefore it MUST NOT contain any information essential to the operation of the PEP. Information that MAY be placed into the Advice Element includes: evidence supporting the assertion, and identification of the policy used in making the assertion.

The Advice element is itself an assertion, or an assertion reference, or any other element from another namespace. An example of how it might be used is as follows. Suppose the assertion authority operates according to a policy uniquely identified by the Object Identifier 1.2.3.4.5.6. (This could be a PKI Certification Authority or Attribute Authority for example). Identification of the governing policy can be provided in the Advice element by setting the namespace to the OID urn of the policy, namely urn:oid:1.2.3.4.5.6Ed Note Not sure this is quite right.

6.2.3 AuthorizationDecisionStatement Element

The AuthorizationDecisionStatement element contains the same elements as the AuthorizationDecisionQuery, and also includes a Decision attribute.

The Decision attribute can take the value of Permit, Deny or Indeterminate. If a value of Indeterminate is returned, then the encapsulating assertion MUST also have a *Conditions* element present expressing the conditions that MUST be fulfilled before the authorization can be permitted[3].

Comment: I don't believe this is correct. A PDP may also return Indeterminate if the resource is outside of it's policy space.

[3] We have to decide on the best way of returning a conditional response. There are a couple of possibilities. I) return Permit with Conditions (but the conditions have to be evaluated to true before the permit is valid) II) return Indeterminate with Conditions (and the decision then depends upon the evaluation of the conditions). II) has been chosen above.

6.2.4 AttributeStatement Element

The AttributeStatement element MAY be included in the Evidence element of an AuthorizationDecisionQuery, to signify the *credential push* mode. For example, when RBAC is being used, the attribute statement could contain the roles of the initiator.

6.2.5 Signature Element

This specification places no constrains on the Signature elements. Implementations SHOULD sign assertions when they do not have an authenticated connection to the evaluator of the assertion.

6.2.6 Required Assertion Fields

Major Revision

MUST be set to 1

Minor Revision

MUST be set to 0

AssertionID

SHOULD be set to a random 128 bit number

Issuer

This SHOULD be the unambiguous name of the issuer. It SHOULD be a URI. Where the Issuer name is an X.500 DN, it MUST have the format as specified in RFC 2255 [RFC 2255]. For example, if the issuer was a PDP with distinguished name of cn=PERMIS ADF, o=University of Michigan, c=us, the URI would be:

ldap:///cn=PERMIS%20ADF,o=University%20of%20Michigan,c=US

IssuerInstant

MUST be the date/time that the Assertion was issued, in ISO 8601 format (i.e. 2003-02-12T12:05:35) and SHOULD be followed by Z to indicate UTC time or the local time zone difference from UTC time.

VW: SAML section 1.2.2 and 2.3.2 states this value must be in the xsd:dateTime format. Unless this is identical to ISO 8601 we're making a serious change here. Why? If it is identical, let's say so.

7. SAML Authorization Service PortType

XXX To be defined

Good start at message below we can leverage:

http://lists.oasis-open.org/archives/security-services/200302/msg00008.html

7.1 Grid Authorization Service SDEs

The following service data elements (SDEs) may be exposed by an Grid Authorization Service.

7.1.1 Supported policies

XXX: A list of policy identifiers that the authorization service knows about.

7.1.2 Policy of PDP in terms of Indeterminate

XXX Whether or not the authorization service supports the Indeterminate response, which some legacy systems may not

7.1.3 Signature Capable

XXX Whether or not the authorization service support signing of its responses.

8. Security Considerations

This specification defines an authorization service based on the SAML specification for OGSA and is completely about security. Implementers of this specification need to take be aware that errors in implementation could lead to denial of service or improper granting of service to unauthorized users.

In particular, mutual authentication between the client and the PDP is highly desirable and strongly recommended. PDP implementations SHOULD sign assertions when they do not have an authenticated connection to the evaluator of the assertion, and MAY sign them when they do have. PDP implementations MAY be unwilling to respond to authorization decision queries from clients who are not authenticated.

Author Information

Von Welch
NCSA
vwelch@ncsa.uiuc.edu

Frank Siebenlist
Argonne National Laboratory
franks@mcs.anl.gov

Sam Meder
University of Chicago
meder@mcs.anl.gov

Laura Pearlman
Information Sciences Institute
University of Southern California
laura@isi.edu

David Chadwick
Information Systems Institute
University of Salford
d.w.Chadwick@salford.ac.uk

Glossary

The following terms are abbreviations are used in this document.

ACI – Access Control Information (from ISO 10181-3). Any information used for access control purposes, including contextual information.

ADF – Access control Decision Function (from ISO 10181-3). A specialized function that makes access control decisions by applying access control policy rules to an access request, ADI (of initiators, targets, access requests, or that retained from prior decisions), and the context in which the access request is made.

ADI – Access control Decision Information (from ISO 10181-3). The portion (possibly all) of the ACI made available to the ADF in making a particular access control decision.

AEF – Access control Enforcement Function (from ISO 10181-3). A specialized function that is part of the access path between an initiator and a target on each access request and enforces the decision made by the ADF.

Client – the entity making a decision request to the ADF (it could be the target, the initiator, or a proxy acting on behalf of the initiator)

Contextual information – Information about or derived from the context in which an access request is made (e.g. time of day).

Environmental parameters – same as contextual information.

Initiator – An entity (e.g. human user or computer-based entity) that attempts to access other entities (from ISO 10181-3).

PDP – same as ADF

PEP – same as AEF

Privilege – An attribute or property assigned to an entity by an authority

Target – An entity, usually a resource, to which access may be attempted (from ISO 10181-3).

Intellectual Property Statement

The GGF 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. 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 implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF 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 practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice

References

[Akenti] Thompson, M., et al., "Certificate-based Access Control for Widely Distributed Resources," in Proc. 8th Usenix Security Symposium. 1999.

[Attributes] Mary Thompson et al. “OGSA Attributes: Requirements, Definitions, and SAML rofile”. GWD-R. Latest version available from https://forge.gridforum.org/projects/ogsa-authz/document/

[Authz] Welch, V., et al, OGSA Authorization Requirments, June, 2003.

[CAS] Pearlman, L., V. Welch, I. Foster, C. Kesselman, S. Tuecke, "A Community Authorization Service for Group Collaboration," Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[OGSI] Foster, I., C. Kesselman, J. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[PERMIS] Chadwick, D.W., O.Otenko, " The PERMIS X.509 Role Based Privilege Management Infrastructure", Proceedings of 7th ACM Symoisium on Access Control Models and Technologies (SACMAT 2002).

[ProxyCerts] XXX

[Roadmap] Siebenlist, F., et al, "OGSA Security Roadmap," OGSA Security WG, Global Grid Forum, July, 2002.

[RFC 2255] T. Howes, M. Smith. "The LDAP URL Format", RFC 2255, Dec 1997

[RFC2904] Vollbrecht, J., et al, " AAA Authorization Framework," RFC 2904, August 2000.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997.

[RFC3281] Farrell, S., Housley, R. “An Internet Attribute Certificate Profile for Authorization”, RFC 3281, May 2002.

[SAML] OASIS, Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1, May 2003.

[SSTC] OASIS Security Services Technical Committee, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security, June, 2003.

[VOMS] "VOMS Architecture v1.1," http://grid-auth.infn.it/docs/VOMS-v1_1.pdf, February 2003.