OASIS eXtensible Access Control Markup Language (XACML)

Technical Committee

Issues List

Version 02

January 21, 2002

Ken Yagen, Editor

Purpose

This document catalogs issues for the eXtensible Access Control Markup Language (XACML) developed the Oasis eXtensible Access Control Markup Language Technical Committee.

Introduction

The issues list presented here documents issues brought up in response to draft documents as well as other issues mentioned on the xacml mailing list, in conference calls, and in other venues. The structure of this document was taken from the Security Assertion Markup Language (SAML) Issues List document maintained at the Security Services Technical Committee document repository. Each issue is formatted as follows:

ISSUE:[Document/Section Abbreviation-Issue Number: Short name] Issue long description. Possible resolutions, with optional editor resolution Decision

The issues are informally grouped according to general areas of concern. For this document, the "Issue Number" is given as "#-##", where the first number is the number of the issue group.

To make reading this document easier, the following convention has been adopted for shading sections in various colors.

Gray is used to indicate issues that were previously closed.

Blue is used to indicate issues that have just been closed in the most recent revision

Yellow is used to indicated issues which have recently been created or modified or are actively being debated.

Other open issues are not marked, i.e. left white.

Issues with lengthy write-ups, that have been closed “for some time” will be removed from this document, in order to reduce its overall size. The headings, a short description and resolution will be retained. All vote summaries from closed issues will also been removed.

Use Case Issues

Group 1: Group Name

Design Issues

Group 1: Group Name

Policy Model Issues

Group 1: Rules

ISSUE:[PM-1-01: Negative Authorizations]

Authorizations can be either positive (permit) or negative (deny). Should we allow both?

See also PM-1-01-A which was split off from this issue.

Potential Resolutions:

[Michiharu] There seems to be agreement on the fact that the core schema should support positive authorizations only. Negative ones are supported as an extension.

[Tim] XACML shall address the requirement for "negative rules" by means of an "and-not-or" construct. [PM-1-01]

[Tim] We use a construct of the following form …

```
<and>
  <rule1/><rule2/><rule3/>
  <not>
    <or>
      <rule4/><rule5/>
</or></not></and>
```

Rule4 and rule5 specify circumstances under which, if either were to hold, access is to be denied. While rule1, rule 2 and rule3 specify circumstances, all of which must hold if access is to be granted.

Champion: Michiharu

Status: Open

ISSUE:[PM-1-01-A: Implementing global deny and Meta-Policies]

Implementing global "deny" semantics using schema 0.8 and meta-policies

[Anne] USE CASE: policy is to deny access to Principal "Anne Anderson" under all conditions. The policy is distributed across many sub-policies, which are all combined to produce the global policy that is to be applied.

Michiharu's concern was with needing to put something like

```
<not><equal>
  <valueRef entity="principal">saml:Subject/NameIdentifier/Name</valueRef>
    <value>"Anne Anderson"</value>
</equal></not>
```

Into every sub-policy if there was no global "deny" syntax.

My proposed solution depends on the idea of having meta-policies. I think meta-policies solve multiple problems:

1. "Where do I get policies",

2. Knowing when you have obtained all the relevant policies,

3. Knowing how to combine policies

4. being able to implement global "deny" and meta-policies does not introduce any new syntax. It is just very explicit in specifying what "applicable policy" means.

Potential Resolutions:

[Anne] Each PDP (or PRP) needs to be configured with a single policy that serves as that PDP's "meta-policy". The syntax of this single policy is exactly that in 0.8.

This "meta-policy" determines where and under what conditions various sub-policies are retrieved. I may not be using <externalFunction> correctly, or the subpolicies may need more enclosing namespace information, but I hope these examples will give the idea. The final example shows how global "deny" semantics are implemented.

EXAMPLE SIMPLE META-POLICY FOR DISTRIBUTED POLICIES:

```
  <?xml version="1.0" encoding="UTF-8"?>
  <applicablePolicy xmlns=...  issuer="<identity that ultimately controls policy for this PDP>"
policyName="...">
    <!-- target omitted, since this policy applies to all targets -->
    <policy>
      <and>
        <externalFunction>http://www.site1/policy1.xml</externalFunction>
        <externalFunction>http://www.site2/policy2.xml</externalFunction>
        ...
      </and>
    </policy>
  </applicablePolicy>
```

What is found at each of the <externalFunction> locations is another <applicablePolicy>, which may be more specific as to which resources it applies to (that applicablePolicy in turn may refer to still other policies). If one of these <applicablePolicy> elements does not apply to the current request, then the result is "does not apply" and does not affect the result of the <and> evaluation.

META-POLICY THAT USES SUB-POLICIES BASED ON RESOURCE

```
<?xml version="1.0" encoding="UTF-8"?>
<applicablePolicy   xmlns=...    issuer="<identity that ultimately controls policy for this PDP>"
 policyName="...">
  <!-- target omitted, since this policy applies to all targets -->
  <policy>
    <or>
      <and>
        <equal>
          <valueRef>saml:Resource</valueRef>
          <value>"file:/host1/*"</value>
        </equal>
        <externalFunction>http://www.site1/policy1.xml</externalFunction>
      </and>
      <and>
        <equal>
          <valueRef>saml:Resource</valueRef>
          <value>"file:/host2/*"</value>
        </equal>
        <externalFunction>http://www.site2/policy2.xml</externalFunction>
      </and>
      ...
   </or>
  </policy>
</applicablePolicy>
```

META-POLICY THAT IMPLEMENTS GLOBAL DENY SEMANTICS

```
<?xml version="1.0" encoding="UTF-8"?>

<applicablePolicy   xmlns=...    issuer="<identity that ultimately controls policy for this PDP>"
 policyName="...">
```

```
    <!-- target omitted, since this policy applies to all targets -->
    <policy>
      <and>
        <not>
          <equal>
            <valueRef entity="principal">saml:Subject/NameIdentifier/Name</valueRef>
            <value>"Anne Anderson"</value>
          </equal>
        </not>
        <or>
          <and>
            <equal>
              <valueRef>saml:Resource</valueRef>
              <value>"file:/host1/*"</value>
            </equal>
            <externalFunction>http://www.site1/policy1.xml</externalFunction>
          </and>
          <and>
            <equal>
              <valueRef>saml:Resource</valueRef>
              <value>"file:/host2/*"</value>
            </equal>
            <externalFunction>http://www.site2/policy2.xml</externalFunction>
          </and>
          ...
        </or>
      </and>
    </policy>
  </applicablePolicy>
```

For administrative ease in a more realistic situation, the set of globally denied attribute/value combinations would be placed in one <externalFunction> policy.

[Ernesto] I support this proposal. I believe it could deal smoothly with the distributed scenario Anne described many times during the last conference call. It goes in the same direction of a previous suggestion of mine (deal with composition and distributed deployment at the ApplicablePolicy level), but does it far better. However, I would suggest some minor observations/amendments (otherwise there is no fun :-))

1. Maybe this is trivial, but any change to the current schema should keep policies fully embeddable in the Applicable policy element, besides being able to point to them using external functions. In simple environments there will be only one local policy, stated in a single document.

2. I happen not to like very much using the word "meta-policy" to describe this proposal, for several reasons some of which would be too long to explain in this message. Basically, I regard Anne's technique mainly as a way to define how a global policy can be deployed in distributed, independently maintained retrieval units. In passing, it also solves the problem of stating which criterion should be applied to compose the outcome of such units (this is essential when "deny" is a possible outcome, as the criterion may have an impact on what actually needs to be retrieved), but I cannot convince myself this requirement is equally important. I believe (but would like to hear the opinion of the industrial researchers on this one) that there will be a default policy composition technique that will be used 99.9% of the times. Therefore, in the schema I would prefer to concentrate the deployment description functionality in a new element, perhaps called "ApplicablePolicies" , possibly defined as an extension of the base (Applicable)Policy type. This element could optionally (via an attribute) specify the composition criterion as well. Tim, what are your views?

[Hal] I am not sure if I agree with Anne's approach. I certainly like it better than the alternative proposed. I actually thought we had previously agreed that there had to be some rules (policy) for determining how independently created policies should be combined to achieve an authorization decision.

Instead of meta-policy, which I think Ernesto fears will be take to mean "more abstract policy" or "policy about policy", perhaps something like Policy Federation Rules would be better.

It seems to me the key issues are:

1. Where and how are PFR specified? Anne's approach is a distinct XML document, which must be consistent throughout the policy federation. This seems reasonable to me.

2. What are the possible PFR's? I think "AND" is impractical, and "OR" is most likely, however some kind of best-match-to-target is conceivable although perhaps too expensive to implement in practice.

3. Do all legal PFR's have to support all decision strategies? I have been thinking about this and I think the right approach is to explicitly call out the possible decision strategies and for each legal PFR state which can or cannot be used.

Here's what I have so far on decision strategies.

Strategy I - Basic

1. Collect all applicable policies
2. Obtain all required inputs
3. Evaluate all policies
4. Apply PFR to resolve conflicting results

Strategy II - Optimized

1. Collect all applicable policies
2. Use PFR to create equivalent combined policy
3. Evaluate policies incrementally, gathering inputs as needed, defer evaluations based on inputs requirements (this for example allows "lazy authentication" where authentication is not done if the result can be determined without it)
4. Once the result is known, stop evaluation

Strategy III- Incremental collection

1. Collect "some" policies
2. Obtain required inputs
3. Evaluate current policy set
4. Use PFR to combine latest results with previous results (if any)
5. If result is known, stop evaluation
6. If not all policies have been collected, repeat previous steps

These are all the possibilities I can think of. Can anyone think of others? I think anything proposed to date works equally for I and II, but not all work for III. However, we may find future possibilities that only work for one of them.

To answer Ernesto's question, our product uses "OR" for authorization decisions and "AND" for audit decisions and there have been no complaints. However we do not have post conditions, which may change things.

As far as the global deny, I would like to understand the requirements better. It seems the problem Anne is trying to solve is "master policy admin can globally deny regardless of what the policy combining rules are."

Is this the right problem to solve? If an "OR" combining rule is used (which I happen to think is the most common case) then any admin can implement a global deny without any special machinery. I think the example given is a red herring to some extent, because the right way to cut off an individual user is to change their attributes at the Attribute Authority or revoke their credentials.

The problem I see is that most evaluation engines will want to use a relatively fixed decision strategy in order to optimize it according to the criteria that apply in that environment. Finding it out in the middle of policy evaluation will interfere with this goal.

[Michiharu] I also support Anne's proposal. I think this technique deal with the distributed scenario nicely. I said the similar idea that uses an external function to call sub applicable policies in the policy model con-call on Dec. 17 but Anne's description is much more concrete and easy to understand. For the global deny policy, I agree that this technique is useful to specify the global deny semantics. If this technique is agreed, we may need more intuitive name for the externalFunction.

[Pierangela] I agree with the fact that the current proposal is able to implement the global deny scenario. No doubt about that: if you restrictions (i.e., the deny you want to enforce) ANDED with the other possible policies nobody will be able to overrule your restrictions.

The reason why I am not too excited with the current proposal is that it seems perfectly fine for communicating policies, but it seems complex to manage.

First of all you have to make sure that the applicable policy is in a single place (sure possibly using URL of other policies) but you cannot allow overlapping targets (which seemed to be the case till now, I believe).

Second the priority of your rules is explicitly managed with the policy definition, which may make administration heavy. Who is in charge of specifying the applicable policy? This will be the only one able to specify global deny: if understand Tim/Anne's proposals correctly possible negative authorizations in other policies have the effect only within that policy (this is fine with me, it seems conceptually clean).

Now for instance, suppose you want to enforce a situation in which any of us can grant authorizations and, possibly denials, for some access and a denial-take-precedence policy should be enforced (meaning it sufficient that one of us says "deny (because of a negative authorization), and the access should be rejected. How do you enforce this? You cannot have the different administrators operate on the applicable policy (meaning actually have writing privilege on that document).

Champion: Anne

Status: Open

ISSUE:[PM-1-02: Post-Conditions]

The current schema [Tim, Jan.3] mentions post-conditions, distinguishing between external and internal, depending on whether their execution requires dialoging with external entities. The current schema suggests (via a comment) that post-conditions can be expressed as invocations of SOAP services. Post-conditions are still to be discussed in details: what is their semantics; how are they executed? A complication of post-conditions associated with a rule involves the distributed scenario (see POLICY COMPOSITION issue). In fact, if I say that a post-condition should be applied whenever a rule fires then I have to evaluate *all* rules. A possible way to overcome this problem is to consider that post-conditions associated with the authorizations that were evaluated to get to an access decision should be executed [Tim]. Note: a possible drawback of this approach is that deterministic behavior may be lost. For instance, there may be N rules applying to an access. If the evaluation of 1 of them brings to a ``permit" decision (so there is no need to evaluate the others). Then, you would ignore the post conditions possibly associated with the other N-1. Different execution of the same request on the same state could then have a different behavior (because a different rule is considered as authorizing the request.

[Tim] The alternative view is that post-conditions must be executed if and only if the associated rule contributes to the permit decision.

[Polar] What is the purpose for actions (i.e. these post conditions) after checking a policy? What types of actions are allowed? Do they change the state of the policy?

[Pierangela] examples that were brought up for post-conditions were things like "logging the request", essentially they are actions that the system executes in response to granting an access, or simply having evaluated the authorizations (discussion on the specific behavior is still open).

Do they change the state of the policy? If you mean the set of rules I guess the answer is no (they should not change the rules). But again, post-conditions are one of the issues which have not discussed fully.

[Polar] Well, I had originally thought that a "post-condition" would be something that would be true if the policy evaluated to true according to its input. That is, a "post-condition" should be a logical consequence, but maybe not fully derivable by all available information. This post-condition would merely be some advice to the evaluator.

Such as Policy stating that:

> Subject is in Role of MissleLauncher to the Resource of Missile on Action Launch.

Post-condition Subject is dangerous.

I really don't like the fact that these post conditions mandate that some generic operation be performed, i.e. it could be used to alter state, especially the state of the policy.

[Simon] Post-condition is executed after the rule fires and does not affect grant/deny

Outcome of the rule. With this definition we can not predict which post condition(s) will be executed for a given authorization request. This is not desirable. One way to make post-conditions predictable is to associate post condition not with a rule but with the outcome of grant or deny, e.g.:

on_grant do_something
on_deny do_something

That means every time any subject is granted (or denied) action on any resource all post-conditions listed in on_grant (or on_deny) will be predictably executed. On_grant and on_deny post-conditions could be associated with specific action, subject, and resource triplet, meaning that given post-condition will be executed every time subject is granted or denied permission to access resource.

```
on_grant(action, subject, resource) do_something;
on_deny(action, subject, resource) do_something;
```

[John]

> Post-condition is executed after the rule fires and does not affect
> grant/deny outcome of the rule.

I thought this was only true of *external* post-conditions? I thought that an internal post-condition must be executed (by the PDP) BEFORE the response is asserted, and therefore does affect the outcome...

The spec says:

"...Post-condition - A process specified in a rule that must be completed in conjunction with access. There are two types of post-condition: an internal post-condition must be executed by the PDP prior to the issuance of a "permit" response, and an external post-condition must be executed by the PEP prior to permitting access..."

I'm assuming that the "musts" here imply that the required actions are successfully executed. Is this not the case?

[Simon] The way I remember post-conditions discussions is that outcome of internal post condition does not affect the outcome of azn decision, i.e., first grant (or deny) is computed and then internal post-condition is executed. If, for example, pdp fails to add a record to the log it still returns computed outcome (grant or deny) to the pep. So the internal post-condition may not be successfully executed by the pdp.

[Tim] This can be accomplished with the current syntax.

```
   applicablePolicy/policy/rule+post-condition
```

This post-condition is executed if access is permitted.

```
   applicablePolicy/policy/not/Rule+post-condition
```

This post-condition is executed if access is denied.

[Bill]

If given this:

> With this definition we can not predict which post condition(s) will be

> executed for a given

> Authorization request. This is not desirable.

'do_something' cannot be guaranteed:

> on_grant(action, subject, resource) do_something;

> on_deny(action, subject, resource) do_something;

Because that would require acknowledgement that it occurred (implying dependence on grant/deny). Sounds like 'post condition' in this sense is more like 'post request'.

[Hal] I clearly remember that the sense of the group was that the PDP MUST insures that an internal post condition occurs, but not necessarily before the permit decision is returned. Post conditions were never considered optional. They are just as required for "permit" as pre-conditions are. That was the rationale for the name.

Potential Resolutions:

[Tim] XACML shall require the PDP/PEP to execute just those post-conditions that accompany the rules that contribute to the "permit" decision. [PM-1-02]

Champion: Simon

Status: Open

ISSUE:[PM-1-03: Post-Conditions as a term]

[Bill] I know that it is late to bring this up, but I find the term 'post condition' unintuitive. Typically, this phrase means the *state* of something after an action, not something to be acted upon. It seems that the way we are using the term implies quite a bit about the context of what is being done. (post what? where?) I think this is being demonstrated by the discussions surrounding the scope of said phrase. In my mind, it would seem that something like 'adjunct policy' or 'adjunct policy condition' would be more appropriate?

[Pierangela] I share this feeling (incidentally, I brought it up in the last conference call, and also in previous once). I was interpreting them more as "actions" than "conditions".

[Pierangela] in today's TC conference call, some people mentioned that "action" is already used with different semantics (=the operation the principal is requesting). That’s true, so we should find another term. The point is, however, that the semantics of "post conditions" now seems really to be a reaction of the system, not the evaluation of a state, so terminology should reflect the semantics.

Potential Resolutions:

1. adjunct policy
2. adjunct policy condition
3. actions

Champion: Bill

Status: Open

Group 2: Applicable Policy

ISSUE:[PM-2-01: Referencing Multiple Policies]

According to the current schema an Applicable Policy seems to refer to a single Policy. The discussions in the last conference call seem to assume that an Applicable Policy can refer to several Policies (distributed scenario and multiple issuers [Anne]). Is there agreement on this point? If so, the schema should be modified accordingly.

Group 1 issues are captured within this

[Tim] The current schema allows one possible way of achieving this. Separate applicable policies from independent PAPs (Policy Administration Points) may be combined in a single "applicable policy" by a PRP. This approach does, however, make the original PAPs anonymous.

Potential Resolutions:

[Tim] An XACML "applicable policy" will not reference external "applicable policies". However, it may "incorporate" external "applicable policies". [PM-2-01] [PM-3-01] [PM-5-03]

[Tim] An XACML "applicable policy" shall be capable of referencing an external "applicable policy", providing explicit rules for combining such policies. [PM-2-01] [PM-3-01] [PM-5-03]

Champion: Anne

Status: Open

ISSUE:[PM-2-02: Target Specification]

According to the current schema each applicable policy can have multiple targets, each of which is an action and a URI identifying a set of resources (possibly with a transfer function to support wildcards). One may want to specify the target with reference to resource attributes (e.g., this policy applies to all files older that two years). How can I specify this?

[Tim] A different transform algorithm is all that is required. In the example, the "classification" is "older than two years", and the transform algorithm specifies how to deduce the age of a file.

Potential Resolutions:

Ernesto suggests that this issue only mention retrieval of distributed policies and should be updated to reflect the recent discussion and Anne's proposal (See PM-1-01A) about policy combination. Anne volunteers to extend its wording in order to include policy combination as well.

Simon will present counter deductions to Anne 's proposal at the F2F

Champion: Simon G.

Status: Open

ISSUE:[PM-2-03: Meaningful Actions]

There are pairings <resource,actions> which are not meaningful (e.g., execute a PDF file) [Simon G.]. Should we control resource/action bindings in the language or refer to an external authority?

Potential Resolutions:

[Tim] The administrative model in Figure 9 deals with this question, placing it out of scope for the schema. If we do need to tackle this, I suggest leaving it for a later version.

[Tim] The XACML syntax shall not address the question of which actions are valid for a particular resource classification. This matter shall be left for implementations to solve in a non-standard way. [PM-2-03]

Champion: Simon G.

Status: Open

ISSUE:[PM-2-04: Indexing Policy]

Also related to target are indexing issues and how to retrieve, given a request, the applicable policy for it [Tim].

Potential Resolutions:

[Tim] Section 6.4 of version 0.8 of the language proposal is reserved for tackling this question in the LDAP case. Do we need to tackle other cases?

[Tim] The XACML specification shall provide normative, but non-mandatory to implement, text that profiles LDAP for distribution of XACML instances. [PM-2-04]

[Tim] The XACML specification shall provide normative, but non-mandatory to implement, text that profiles "the Web" for distribution of XACML instances. [PM-2-04]

Champion: Tim

Status: Open

ISSUE:[PM-2-05: Ensuring Completeness]

The applicable policy is defined as the ``complete" set of policies that apply to a resource. How do I ensure completeness (meaning no two targets should intersect?)

Potential Resolutions:

[Tim] This is a job for the PRP and should (I think) be out of the scope for our specification. The PRP has to be configured with the names and locations of the PAPs whose policies it recognizes.

[Tim] The XACML syntax shall not address the question of ensuring that "applicable policy" is complete. This matter shall be left for PRP implementations to solve in a non-standard way. [PM-2-05]

Champion: Pierangela

Status: Open

ISSUE:[PM-2-06:Policy Security]

Resolution 4: An XACML "applicable policy" will contain its own security features (e.g. signature), rather than relying on an encapsulating saml assertion.

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-2-07: valueRef type]

Resolution 5: XACML valueRef elements shall be of type "saml:AttributeValueType".

Potential Resolutions:

???

Champion: Tim

Status: Open

Group 3: Policy Composition

Assuming an Applicable Policy can refer to several Policy elements, we need to answer the following questions:

ISSUE:[PM-3-01: Combining Policy Elements]

How are the Policy Element combined? For instance, we could support Boolean expressions of policies. E.g., if there are three policies by independent issuers, I can say ``P1 AND (P2 OR P3)? This could fit well in the multiple issuers scenario Anne was envisioning. Should this be part of the core of the extension (external URI [Michiharu])?

Potential Resolutions:

[Tim] We could add "policy" to the "sequence" in "rule". Then we would have to give policies unique identifiers, not just string names. Perhaps, we should add "applicable policy", instead of "policy".

[Tim] An XACML "applicable policy" will not reference external "applicable policies". However, it may "incorporate" external "applicable policies". [PM-2-01] [PM-3-01] [PM-5-03]

[Tim] An XACML "applicable policy" shall be capable of referencing an external "applicable policy", providing explicit rules for combining such policies. [PM-2-01] [PM-3-01] [PM-5-03]

Champion: Michiharu

Status: Open

ISSUE:[PM-3-02: Specifying Policy Outcome]

How the policy outcome should be specified. Possibilities are 2-valued (access decision is ``grant"/"deny") or 3-valued (policy outcome is ``grant"/"deny"/nothing). Note the ``nothing" means that no rule applies, to be solved according to default. (Related work on composition…?)

How does the PEP interpret the answer I don't know?

Potential Resolutions:

[Tim] Ultimately, the PEP has to know whether or not to grant access. So, someone has to decide, and (by definition) it is the PDP. So, the "don't care" response isn't helpful. However, saml should have an error code to indicate that the PDP is not the appropriate PDP to render a decision on a particular request.

[Tim] The XACML specification shall specify when a PDP should return saml:decision attributes with the values "permit" and "deny". If the PDP is unable to render a decision, then a saml status code shall be returned. No decision value shall be supplied in this case. [PM-3-02]

Champion: Simon

Status: Open

Group 4: Syntax

ISSUE:[PM-4-01: Syntactic Sugar]

The current schema assumes authorizations are specified as a pre-condition which is an expression made of predicates on SAML attributes (conditions on principal, resource and environment can be interspersed), let's call it Option ``pre-cond" [Carlisle, Tim, Anne, ...]. In the last conference call it was agreed to leave as an open issue whether to group conditions about principal, resource, and environment in three different elements, let's call it Option ``triplet" [Michiharu, Ernesto, Simon, ...]. The argument for Option ``pre-cond" is that there are predicates that involve both principal and resource attributes (e.g., an authorization that states that users can read the files they own). The counter-objection to this is that you can naturally include all predicates on resources in the resource condition element (which can also refer to principal attributes). The argument for the triplet is that it makes authorization specifications conceptually clearer and closer to current approaches.

[Tim] In the 0.8 schema, valueRef has an attribute to indicate the entity to which it applies (principal, resource, etc.). It only has to be consulted if the attribute type identifier is ambiguous.

Potential Resolutions:

[Tim] The XACML syntax will differentiate between model entities (principal, resource, etc.) in its attribute elements, rather than in its rule elements. [PM-4-01]

Champion: Pierangela

Status: Open

ISSUE:[PM-4-02: Policy names as URIs]

Policy names are strings. Should we make then URIs?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-03: Required type in policy]

The "rec:patient/patientName" element is a complex type. So, how should we indicate the required type in the policy?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-04:syntax extension]

Issue: should this element be an extension point to which other policy syntaxes can be added?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-05:Policy Name a URI]

Issue: should we make policy name a URI?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-06:Comment element]

Issue: Should we include a "comment" element?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-07:policy element in a rule]

Issue: Should we allow a policy element in a rule? Then the same schema could express the policy for combining policies. If so, should it be policy or applicable policy?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-08:XML elements include xsi:type]

Issue: Should we require XML elements compared in this way to include an xsi:type attribute?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-09:complex types]

Issue: This only allows for simple types. Do we need to support values of complex type?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-10:preserve PAP identity]

Issue: Should the identities and/or signatures of the PAPs be preserved in the composed policy?

Potential Resolutions:

???

Champion: Tim

Status: Open

Group 5: SAML Related

In the current schema attributes on resources and principals, which can be used in the Target (for resources) and in predicates, are retrieved using URIs pointing to SAML dataflow.

ISSUE:[PM-5-01: Non-SAML Input]

Can this mechanism be extended to point to non-SAML authorities as required in the Java environment [Sehkar]?

At a minimum, extending SAML expressions but broader to other authorities.

Potential Resolutions:

[Tim] The XACML specification shall be closely coupled to saml entities. However, the use of saml namespace identifiers is not intended to imply that all attributes must be retrieved from saml messages and assertions. [PM-5-01]

Champion: Sehkar

Status: Open

ISSUE:[PM-5-02: Wildcards on Resource Hierarchies]

How do we express wildcards on the resource hierarchies [Simon G.]?

Potential Resolutions:

The current schema includes ResourcetoClassificationTransform to this purpose. Is this sufficient?

[Tim] We should register an OASIS identifier for the use of regular expressions in this context.

[Tim] The XACML syntax shall use registered URIs to identify algorithms for processing resource classification wildcards. [PM-5-02]

Champion: Simon G.

Status: Open

ISSUE:[PM-5-03: Roles and Group Hierarchies]

Are roles and groups hierarchies available via SAML [Simon G.]? Hierarchies could be needed, in case of support of negative rules, for resolving conflicts based on more-specific-takes-precedence. Note: policy resolution conflicts fit well when the principal is a group, they may be difficult to apply in case of principal's expressions.

Potential Resolutions:

[Tim] An XACML "applicable policy" will not reference external "applicable policies". However, it may "incorporate" external "applicable policies". [PM-2-01] [PM-3-01] [PM-5-03]

[Tim] An XACML "applicable policy" shall be capable of referencing an external "applicable policy", providing explicit rules for combining such policies. [PM-2-01] [PM-3-01] [PM-5-03]

Champion: Simon G.

Status: Open

ISSUE:[PM-5-04: SAML Assertions URI]

From the schema it seems that expressions are predicates whose arguments are always URI or value. Are SAML assertions always URI?

Potential Resolutions:

[Tim] Attributes in saml assertions are identified by a namespace, which is a URI, and a name, which is a string.

Simon suggests that the current solution in general enough, as the URI+XPath combination specifies a schema (via the URI) and allows to retrieve a value (via the XPath). XPaths guarantee that values are uniquely identified. This technique smoothly applies not only to SAML but also to other formats like LDAP.

Hal observes that this is not always the case, as there may be attribute namespaces which are not URI.

Anne remarks that besides a pointer to the schema, a pointer to an instance is also needed. Simon agrees to provide a full explanation of this scenario at the F2F.

Champion: Simon

Status: Open

ISSUE:[PM-5-05: XPath]

Use of Xpath for identifying SAML constructs and the use of Xpath operators

Potential Resolutions:

Simon clarifies that the position he will take is that while the use of Xpaths to extract nodeset is just fine, they do not make good values in expression. The solution in the current schema is cleaner.

Anne offers to look into the issue to provide an alternative point of view.

Champion: Simon

Status: Open

ISSUE:[PM-5-06: Multiple actions in single request]

In the SAML issues document, http://www.oasis-open.org/committees/security/docs/draft-sstc-core-discussion-01.doc

... Issue 5.1.15.2 seeks guidance on whether multiple "actions" can be specified in a single decision request.

Potential Resolutions:

[Tim] I feel that XACML should answer this question and send its conclusion in a liaison to SAML. My feeling is that the answer is "No". If "applicable policy" is to be identified with the resource/action pair, then multiple "applicable policies" are involved when multiple actions are involved. Much "cleaner" for there to be a single "applicable policy" for each decision request. And, therefore, a single action per decision request. It is no great hardship to submit multiple decision requests, in the event that you need a decision for each of several actions.

[Hal] Personally I am in favor of limiting this, but I will state the counter argument for the record. If the possible Actions correspond to what can be in the request, then this works fine. The only reason for multiple actions would be some sort of policy provisioning requirement. However, if the Actions are more like privileges or permission bits, and do not match allowable requests one for one, then some requests may require the AND or OR of several actions. I believe this is the motive behind suggesting multiple actions.

I don't see any rush on this as we are not close to proposing changes to the decision protocol yet.

Champion: Tim

Status: Open

ISSUE:[PM-5-07: Delegation]

[Polar] Has anybody thought about how delegation can be reasoned about in XACML? It appears that SAML only asserts a flat list of attributes with a single principal, or am I off base here? Can I support policies on such operations as:

Paul for Peter says debit Peter's account?

Which mean that Paul (or some other party trusted to do so) has issued Paul the authorization to act on behalf of Peter, in this case to access Peter's account. Or such things, like WebServer quoting JohnDoe says lookup in customer database. Where the WebServer may be trusted to authenticate JohnDoe, but no such proof is necessary other than the WebServer merely claiming to be acting on JohnDoe's behalf?

Potential Resolutions:

[Hal] With regards to SAML, the Access Decision Request was deliberately kept simple with the idea that XACML would give us the tools to do the job properly. I have proposed (see my use cases) that XACML not only be able to express policies, but the method of expressing policy inputs be rolled back into the SAML Access Decision Request (and Assertion).

In my opinion, XACML policies should be able to contain predicates about zero or more of the following subjects:

Requestor Subject

Recipient Subject (can be different from requestor)

Intermediary Subject (can be more than one for a given request)

I propose a single construct for Subjects and their attributes and some kind of modifier indicating the type (refrain from using "role" here) of subject.

[Tim] Delegation could be expressed in attribute assertions. The very issuance of an attribute assertion is a form of delegation. So, XACML should not have to concern itself with the process by which an entity obtained an attribute.

Champion: Polar/Hal

Status: Open

ISSUE:[PM-5-08: saml;Action is a “string”]

These are some of the potential SAML issues. Most of them were found when attempting to write J2SE policy files in XACML syntax. Further discussion is needed on these issues.

saml:Action is currently specified as a "string". Making Action an abstract type would allow it to be extended. This would allow the content model to be defined by a schema external to the SAML spec.

Thus what constitutes an action could be determined by the J2SE schema.

Potential Resolutions:

[Toshi] In SAML, saml:Action is used only in saml:Actions and saml:Actions have Namespace as an attribute. So it is possible to write action(s) such as:

```
<saml:Actions Namespace="urn:J2SEPermission:java.io.FilePermission">
    <saml:Action>write</saml:Action>
</saml:Actions>
```

or

```
<saml:Actions Namespace="urn:J2SEPermission">
   <saml:Action>java.io.FilePermission:write</saml:Action>
</saml:Actions>
```

But it will be useful if we can write something like:

```
<saml:Action>
    <J2SEPermission class="java.io.FilePermission">write</J2SEPermission>
</saml:Action>
```

Champion: Sekhar

Status: Open

ISSUE:[PM-5-09: saml;AuthorizationQuery requires actions]

If actions are optional for XACML, then why should <saml:Actions> be required in <saml:AuthorizationQuery> ? Both the wording in the SAML assertions draft as well as the SAML schema places such a requirement. saml:Actions should be optional in the AuthorizationQuery to accommodate queries without actions. At least for now, I don't anticipate this as an issue for J2SE.

Potential Resolutions:

[Toshi] In the latest SAML spec (core-25), AuthorizationDecisionQuery element has Resource attribute and Actions element and both of them are "required". Does this cause many problems?

(Resource attribute is "optional" for AuthorizationDecisionStatement element.)

As for J2SE case, I think there is an issue in terminology.

Champion: Sekhar

Status: Open

ISSUE:[PM-5-10: single subject in AuthorizationQuery]

[editor note: Is this issue covered somewhere else?]

saml:AuthorizationQuery currently only contains a single Subject. While a saml:Subject can support multiple NameIdentifier or SubjectConfirmation or AssertionSpecifier elements, it is required that they all belong to the same principal. So a single subject cannot be used for unrelated principals. In J2SE, there is a need to base access control on multiple principals which are not related and this therefore points to a need for more than one Subject in the saml:AuthorizationQuery

Potential Resolutions:

The way out of this appears to be extend SubjectQueryAbstractType.

Champion: Hal

Status: Open

ISSUE:[PM-5-11:XACML container in SAML]

Issue: should we use a SAML assertion as a container for an XACML applicable policy?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-5-12:derive attribute from saml:AttributeValueType]

Issue: Should we derive the attribute from saml:AttributeValueType? This seems to make sense, but the resulting attribute will have to become an element, with start and stop tags, making it larger and less readable.

Potential Resolutions:

???

Champion: Tim

Status: Open

Group 6: Predicate Cononicalization

ISSUE:[PM-6-01: SAML Assertions URI]

Values used in predicates can refer to various standard formats (e.g, X.509 [Anne]) that could make the predicates evaluation difficult. For instance, if a principal's name is expressed in X.500

syntax you cannot compare it against a simple string. How do we make the representations canonical?

Potential Resolutions:

[Tim] Policy environments have to use consistent type definitions for the attributes they use.

Champion: Anne

Status: Open

Group 7: Extensibility

ISSUE:[PM-7-01: XACML extensions]

XACML Extension Model that defines what portion of the XACML specification is a core and to what extent the XACML specification can be extended. Based on this proposal, XACML policy administrators can represent much broader access control policies by extending the core portion of the XACML specification.

This extension model is designed to support an XACML extensibility property stated in the XACML charter. This proposal is based on the current language proposal document but includes several modifications.

Potential Resolutions:

See http://lists.oasis-open.org/archives/xacml/200112/msg00076.html

Champion: Michiharu

Status: Open

Miscellaneous Issues

Group 1: Glossary

ISSUE:[MI-1-01: Consistency]

Pierangela mentioned something discussed in PM group that may not coincide with glossary concerning pre and post conditions.

Potential Resolutions:

???

Champion: Pierangela

Status: Open

Group 2: Conformance

ISSUE:[MI-2-01: Successfully Using]

XACML definition of OASIS requirement to successfully use the specification

Potential Resolutions:

"Successfully Using the XACML Specification"

XACML is an XML schema for representing authorization and entitlement policies. However, it is important to note that a compliant Policy Decision Point (PDP) may choose an entirely different representation for its internal evaluation and decision-making processes. That is, it is entirely permissible for XACML to be regarded simply as a policy interchange format, with any given implementation translating the XACML policy to its own local/native/proprietary/alternate policy language sometime prior to evaluation.

A set of test cases (each test case consisting of a specific XACML policy instance, along with all relevant inputs to the policy decision and the corresponding PDP output decision) will be devised and included on the XACML Web site.

In order to be "successfully using the XACML specification", an implementation MUST, for each test case, have a "policy evaluation component" that can consume the policy instance and the inputs and produce the specified output.

Furthermore, the implementation MUST have a "policy creation component" that allows it to generate schema-valid XACML policy instances that can be consumed/processed by other PDPs.

Note that, aside from the XACML policy instance itself, all PDP inputs and outputs MUST be SAML-compliant (i.e., conform with the assertions and protocol messages defined in the SS-TC SAML specification), although other syntaxes/formats for the PDP input and output MAY be supported in addition to this.

Champion: Carlisle

Status: Closed

Group 3: Patents, IP

ISSUE:[MI-3-01: XrML]

[Ernesto] As I recollect, OASIS requested us to evaluate whether any XACML specification might fall in the scope of patents held by others. I quote from a Dec 13th addition to announcements regarding Xerox's XrML:

(http://xml.coverpages.org/xrml.html) :

"ContentGuard's strategy appears to be to make money by licensing the technology -- whatever some outside body defines it to be. It can do this because its patents cover the idea of a rights language in general, no matter what the specifics of the language are".

I know XrML has already been mentioned in our discussions from the technical point of view, but the wording of this announcements makes me suspect that we should explore the matter further from the patents' point of view.

Potential Resolutions:

Oasis has a specific IPR policy and ContentGuard needs to make Oasis aware of any IP as it relates to XACML or other technical committees in accordance with that policy.

[Hal] Paragraph (C) of OASIS.IPR.3.2. makes the following points:

If OASIS knows about something they "shall attempt to obtain from the claimant of such rights a written assurance ..."

However, "results of this procedure shall not affect advancement of a specification..."

Except that "The results will, however, be recorded..." and "...may also direct that a summary of the results be included in any OASIS document published containing the specification." It also says elsewhere that they will not go out of their way to find IPR that has not been drawn to their attention.

Champion: Ernesto

Status: Open

Group 4: Other Standards

ISSUE:[MI-4-01: RuleML]

Should XACML look at RuleML?

[Edwin] XACML folks, Since XACML is about defining "rules" for Authorization -- would it make sense to leverage work done by the RuleML folks?

RuleML folks, You may want to checkout XACML as an application of RuleML. Here is a standard that will be real within the next year!]

Potential Resolutions:

The issue is a generic suggestion about XACML to be a possible application of a general setting for rule representation, RuleML.

Anne proposes that at the F2F every suggestion of taking into account related languages should be mandatory accompanied by a presentation

After a brief discussion on RuleML, the issue is voted closed. It should be deleted from the next version of the issues document

Champion: Edwin

Status: Closed

ISSUE:[MI-4-02: RAD]

Should XACML look at RAD?

[Polar] In response to some query about the expressiveness of evaluation of policies from different places, I would like to point the group to the CORBA Resource Access Decision specification (RAD).

http://www.omg.org/cgi-bin/doc?formal/01-04-11.pdf

and we may want to include it the document repository. It has in it an Access Decision model in which not only policies are located, but also, a policy evaluation combinator is located for a particular resource. Note, there is no language component to this specification.

However, it does present a model by which policy can be distributed and evaluated. A combinator, which has an interface operation of "evaluate_policies" takes the list of located policies for the resource, the attribute list of the subject, and the operation (i.e. Action) on the resource) and evaluates the decision.

That way, depending the semantics of the combinator you choose for the resource, your combinator may choose to ignore, or evaluate only some policies based on the evaluations of other policies.

Potential Resolutions:

Polar will bring that one to the discussion, with special reference to policy combination.

Champion: Polar

Status: Open

ISSUE:[MI-4-03: DSML]

Transformations from XACML to DSML

[Gil] Since the last time we talked I had the chance to play with DSML a little. It seems to me that it is theoretically possible to transform an XACML policy document into a DSML document and import that document into LDAP. The DSML document could contain elements that described the (LDAP) schema necessary to store the authorization policy entries in case the target LDAP

didn't already have this schema. It is also possible to export some LDAP entries into a DSML document and transform that DSML document in XACML.

What I don't know (having nothing more than a cursory understanding of XSL/XSLT) is how difficult such transformations would be and if there are any "gotchas" that would keep this from really working.

Potential Resolutions:

[Gil] What I think the XACML spec should do is:

1.) Describe the LDAP schema necessary to store authorization policies. This should be done in "LDAP fashion" with dn's, classnames, etc.

2.) (if possible) Provide the XSLT necessary to transform XACML to DSML and vice versa.

That way people who don't want to be bothered with DSML can work out their own way to store and retrieve XACML data to and from the defined schema.

Champion: Gil

Status: Open

ISSUE:[MI-4-04: Java Security Model]

Hal says he is not clear about whether XACML should be able to represent the Java security model. Gil comments that XACML would be limited if it cannot express it. Hal notes that what XACML should be able to represent are the same requirements that Java security model represents, but not necessarily in the same way (i.e., representing the same authorizations).

Potential Resolutions:

???

Champion: Sekhar

Status: Open

Document History

- 7 Jan 2002 First Version Published

- 21 Jan 2002 Major edits and additions. Every open item updated.