IDS Authorization Problem Statement

June 8 2010

R. Turner

Introduction

This memo briefly introduces the notion of authorization as it applies to hardcopy devices. There are many authorization problem scenarios that could be addressed, however this memo only describes a few of these scenarios. This memo also introduces the concept of authorization policies, as well as a potential way of expressing authorization policies.

Review of Generalized Authorization Problem

Authorization, as implemented in most existing systems, tries to answer the general question:

Who can access what, under what conditions, and for what purpose ?

This same general question can also be applied to use-cases (or "user stories") describing how end users interoperate with hardcopy devices.

In order to answer the question (or "problem") laid out above, I have identified (in "red") the information that must be ascertained prior to answering the question.

For example, the "Who" is normally derived through some type of prior authentication process. This memo correlates the "Who" with an "identity". The "what" is normally the target of an operation being attempted by the "Who". The "conditions" can be abstract or concrete, and describe one or more variants that have a "temporal" quality. The "purpose" is usually self-evident, and is implied by the operation. But their are cases where the "purpose" of the access can have many permutations. For example, if a "Who" is accessing a "printer", the purpose is usually to print a document of some kind. However, there are exception cases where the purpose might be to "manage" the printer in some way; not printing a document at all.

You can think of the who, what, conditions, and purpose as terms of an equation that must be known prior to evaluating a result.

All of these terms makeup “predicates” to the overall authorization problem (i.e., the result of the authorization equation is “predicated” on the values of these terms).

Also, in general, the answer to the authorization equation is either a “yes” or “no”; either allow access or deny access, based on the predicates.

What makes the problem even more difficult is expressing complex (often “compound”) forms of the authorization question.

For example, a simple expression of the authorization problem in a hardcopy use-case might be:

“John” can access “Printer-A” under the following condition: “If the current time is between 9 AM and 5 PM” for the purpose of “printing”.

However, the “conditions” can be compounded, and other terms could be multi-valued, as in:

“John” can access “Printer-A” under the following conditions: “If the current day is a weekday” OR “the current day is Saturday” OR “the current day is Sunday” AND “the current time is between 9 AM and 5 PM”

We’re basically trying to say that John can use the printer anytime during the week, but if it’s the weekend, he can only use the printer between the hours of 9 AM to 5 PM.

The above simple and complex predicates combine to form a “policy” statement. In most systems, policy statements can be compounded, and the compound policy statements can be combined. In other words, policies can be quite difficult to express in some cases. And these policies can especially be difficult to express in a way that is understandable (and enforceable) by an automaton (software).

Hardcopy Device Authorization

One of the work items that the IDS group can consider is defining the possible values for the “what” and “purpose” predicates discussed in the previous section.

I think there are already mechanisms in place to robustly establish the “who” predicate, using authentication methods already employed by most hardcopy device vendors. The only challenge is making sure that, if a vendor utilizes multiple authentication methods for a particular device, all of the authentication methods must somehow map to a common “identity”.

The “what” predicate would map to some “resource” in the device, and the “purpose” would map to some operation or function to be performed on the resource, or using the resource.

Examples of the “what” predicate for hardcopy devices might include:

- Particular input bins
- Particular output bins
- Color/Ink resources
- fax subsystem
- printing subsystem
- scanning subsystem
- System configuration parameters
    - defaults
- other

Examples of the “purpose” predicate for hardcopy devices might include:

- printing
- faxing
- scanning
- readonly-management
- read/write management
- Maintenance
- other?

Example “conditions” for a hardcopy devices could overlap with general authorization conditions used in other applications, such as:

- particular date/time ranges
- Day(s) of the week
- User attributes
- other?

XACML

In simple, single-function hardcopy devices, authorization policies may be very simple; whereas with a high-end, high-duty-cycle, multi-function device, authorization policies could be compound, and quite complex. Given the wide range of policy expression that may be required, I would like to suggest that we use XACML as an XML dialect for expressing policies based on predicates.

The eXtensible Access Control Markup Language (XACML) is an OASIS Standard that provides:

- Policy Language
- Request and Response Language
- Standard data-types, functions, combining algorithms
- Extensibility
- Privacy profile and RBAC profile

Not all of these features may be required for an authorization implementation. However, for hardcopy device authorization use-cases, it is assumed that the Policy Language, Standard datatypes/functions and combining algorithms would be particularly useful. XACML can also be used to express legacy “role-based” (RBAC) models to support transition of legacy authorization systems to a single, more robust policy specification.

XACML defines the following terms that are used in the XML dialect for XACML:

Resource - Data, system component or service

Subject - An actor who makes a request to access certain Resources.

Action - An operation on a resource

Environment - The set of attributes that are relevant to an authorization decision and are independent of a particular subject, resource or action

Attributes - Characteristics of a subject, resource, action or environment

Target - Defines conditions that determine whether policy applies to request

From the above list of terms, you can start to see how the traditional authorization predicates are mapped to XACML equivalents.

The model for XACML works with Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs), much in the same way that we have seen previously with NAC/NAP implementations. The XACML model extends the model to define a Policy Information Point (PIP) that serves as the source of attribute values, or the data required for policy evaluation.

The XACML model is a request/response model wherein a subject requests access to a resource, including how it intends to use the resource, and a response is returned indicating whether to “permit” or “deny” the access.

XACML Request Sample

```
<Request>
    <Subject>
              <Attribute AttributeId="urn:oasis:names:tc:xacml:
              1.0:subject:subject-id"
              DataType="urn:oasis:names:tc:xacml:1.0:data-
              type:rfc822Name">
              <AttributeValue>xyz@users.example.com</AttributeValue>

              </Attribute>
               <Attribute AttributeId="group"  DataType="http://www.w3.org/
    2001/XMLSchema#string”
              Issuer="admin@users.example.com">
              <AttributeValue>developers</AttributeValue>
              </Attribute>
    </Subject>
    <Resource>
              <Attribute AttributeId="urn:oasis:names:tc:xacml:
              1.0:resource:resource-id" DataType="http://www.w3.org/2001/
    XMLSchema#anyURI">
                     <AttributeValue>http://server.example.com/code/docs/
    developer-guide.html
                     </AttributeValue>
              </Attribute>
    </Resource>
    <Action>
              <Attribute AttributeId="urn:oasis:names:tc:xacml:
    1.0:action:action-id" DataType="http://www.w3.org/2001/
    XMLSchema#string">
              <AttributeValue>read</AttributeValue>
              </Attribute>
    </Action>
</Request>
```

XACML Response Sample

```
<Response>
    <Result>
              <Decision>Permit</Decision>
              <Status>
              <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
              </Status>
    </Result>
</Response>
```

Effect:
Permit/Deny/Not Applicable/Indeterminate

The open source Sun XACML project and the Apache Axis2 project utilize XACML for authorization, and these systems have been deployed in larger identity management and authorization systems that federate user authorization across authentication domains (SAML).

References

More information on XACML can be found at the following links:

OASIS XACML Technical Committee Home Page
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

Sun's XACML Open Source Implementation
http://sunxacml.sourceforge.net/