Identity, Privacy, and Data Protection in the Cloud – XACML

David Brossard
Product Manager, Axiomatics
What you will learn

- The issue with authorization in the cloud
- Quick background on XACML
- 3 strategies to extend authorization to the Cloud
- What it means for
  - customers
  - SaaS providers
The issue with Authorization today

The black box challenge
System growth leads to AuthZ challenges

- Cost
- Brittleness
- Static
- Risk
- Lack of visibility
- Lack of audit
- Violation of SoD
What happens to my data?

Who can access which information?

How do I comply with (what the auditor will ask for)

- Regulations?
  - E.g. Export Control

- Contractual obligations?

Going to the cloud doesn’t make it easier

- Do I need a different approach for cloud?
Example: Manufacturing in the cloud

Export Control

- Know the user (citizenship, location, affiliation)
- Know the end use (end location, purpose of use)
XACML to the rescue

Implementing fine-grained authorization in the cloud
Authorization is nearly always about

Who?

Identity + role (+ group)

Role-based Access Control
Authorization should really be about…

Attribute-based Access Control

Behold XACML, the standard for ABAC

- eXtensible _Access _Control _Markup _Language
- OASIS standard
- XACML is expressed as
  - A specification document (a PDF) + XML schema
- Policy-based & attribute-based language
  - Implement authorization based on object relations
  - Only employees of a given plant can see technical data linked to items assigned to the plant
Who’s behind XACML?

- Oracle
- IBM
- Veterans Administration
- Axiomatics
- $\text{EMC}^2$
- Bank of America
- The Boeing Company
- And many more...
Refresher: the XACML architecture

- Enforce: Policy Enforcement Point
- Decide: Policy Decision Point
- Support: Policy Information Point, Policy Retrieval Point
- Manage: Policy Administration Point
Centrally managed policy:
"PERMIT user with clearance X to read document classified as ..."
"DENY access to classified document if..."
XACML ➞ Anywhere AuthZ & Architecture

Datacenter
- App A
- Service A

Private Cloud
- Service A

SaaS
- Service D
- Service E

SaaS
- Service M
- Service O

PEP ➝ PDP

OASIS
Fine-grained Authorization for the Cloud

Three strategies for externalized authorization in the cloud
A SaaS provider should offer
- Functional APIs (their core business)
- Non-functional (Security) APIs

Let customers push their own XACML policies

Apply the administrative delegation profile

http://docs.oasis-open.org/xacml/3.0/xacml-3.0-administration-v1-spec-en.html
Option #1 – Architecture

1. *SaaS Admin delegates rights to manage access control provided to customer A. The rights are restricted to only the applications and resources provided to this particular customer’s users.*

2. *Customer A’s admin can manage access for their staff on its own by providing XACML policies and attributes*

3. *Customer A users use the SaaS application*
Option #1 – Architecture (including id. Federation)

1. User authenticates using internal authentication mechanism and id store (e.g. LDAP)
2. User calls out to SaaS application
3. User internal identity is translated to virtual identity e.g. SAML assertion
4. Internal user identity is verified. User attributes are inserted inside newly created SAML assertion
5. User message is forwarded to SaaS app along with SAML assertion
6. SAML assertion is validated against local STS. Attributes are extracted from SAML assertion and returned
7. XACML AuthZ request including the attributes from the SAML assertion are sent to the PDP for a decision
Option #1 – Pros & Cons

**Pros**
- Consistent access control
- Fine-grained
- Risk-aware
- Future-proof
- SaaS vendor benefit
  - multi-tenancy

**Cons**
- Not many SaaS vendors support XACML today
If you can restrict access to SaaS applications from within the corporate network...

All access to SaaS apps could be made to tunnel through a proxy
Option #2 – Architecture

VPN

PEP

PDP

SaaS App #1

SaaS App #2

SaaS App #3
Option #2 – Pros & Cons

**Pros**
- Workaround current SaaS limitations
- Easy to deploy
- Available today

**Cons**
- No direct access to SaaS app
  - Forces users to go via VPN
- Access may not be as fine grained as Option #1
  - Lack of visibility into the SaaS data
What if the provider is reluctant to adopt XACML?

“If the application won’t go to XACML then XACML will go to the application”

Eve Maler, Forrester

You still get

- Centrally managed authorization
- Standards-based (XACML)

Approach

- Convert from XACML to expected SaaS format
- Push via SaaS management APIs
Option #3 – Architecture

Convert XACML policies to the native format expected by the SaaS provider

Authorization constraints / permissions in the format expected by the SaaS provider

Customer A users use the SaaS application

Company A

SaaS provider

App#1

App#2

App#3

Native API

Functional API
Option #3 – Pros & Cons

Pros
- Feasible today
- Viable solution
- Extends the customer’s XACML-based authorization system’s reach

Cons
- Possible loss of XACML richness in access control
- Loss of dynamic nature
Standards & the Cloud

Standards are important for the cloud

- It promotes vendor interoperability
- It promotes layer interoperability

Example

- XACML authorization services can easily use SAML, OAuth, OAuth2, OpenID...
- XACML can also use semantic web standards

This leads to easier deployments and faster ROI
To summarize

- Cloud requires eXtensible Authorization
  - Fine-grained
  - Externalized

- Traditional approaches
  - #1: tell your SaaS provider to adopt XACML.
  - #2: proxy your cloud connections.

- Extended approach
  - #3: Policy Provisioning based on XACML
  - Also works for business apps (SharePoint, Windows)

Every cloud has a XACML lining
Online resources

- LinkedIn group: [http://www.linkedin.com/groups/OASIS-XACML-3934718](http://www.linkedin.com/groups/OASIS-XACML-3934718)