Beyond Access

Using ABAC frameworks to implement privacy and security policies

AUDIENCE
Usenix PEPR '20

AUTHOR
Amanda Walker
Nuna, Inc.

DATE
10-15-20

A brief history of access control

Physical

Access is controlled by a physical attribute of a device or piece of physical media..

Example: write-protect tabs on a floppy disk

Identity

Access is a function of who is requesting access to a resource.

Example: UNIX® permissions, traditional ACLs.

```
-rwxr-x---
```

Role

Access is a function of the user's role(s), and authorization associated with those roles.

Examples: RBAC, group membership tests

Context

Access is a function of attributes of the user, the data, source and destinations, the nature or purpose of a computation, and other policy constraints.

Attribute Based Access Control

- **First appeared in 2000**

- **Multiple standards!**
  - OASIS XACML (2001)
  - NIST SP 800-162 (2014)
  - Microsoft SDDL
  - … and so forth

- **Common groups of attributes:**
  - Subject
  - Action
  - Object
  - Context

Two general approaches

Object attributes

Pros:

- Fast: rules can be evaluated inline
- Transparent: rule is explicit at the policy enforcement point

Cons:

- Decentralized: rule updates can be painful
- Not every useful attribute is static

Policy service

Pros:

- Centralized: rules can be updated and take effect immediately
- Dynamic attributes can be computed or looked up on demand

Cons:

- Much slower
- Service must be reachable

Access is a special case

... of "should this computation proceed?"
... or "should this computation include this data?"

Purpose

User interaction
Personalization
Monetization
Research
Security / anti-abuse
Ancilliary uses

Jurisdiction

Different countries have different rules

Even jurisdiction is a function, not a static attribute

Public Policy

"Public" does not mean unconstrained (see RTBF)

Users have expectations even around "public" information about them

Internal Policy

"Keeping honest people honest"

Think beyond access

- Many privacy policies are not about who can access what, they are about purposes and jurisdiction
- Expectations, agreements, and regulations change. A layer of indirection makes it much easier to adapt.
- Ask why: access rules (especially RBAC) are often proxies for more abstract rules that can be computed dynamically
- If you can't write a piece of code to evaluate a policy, ask:
    - What information is missing?
    - Can it be recorded somewhere a policy service can read it?
    - Do other systems expose state that affects whether a policy applies?

Thank You.

Amanda Walker VP, Engineering awalker@nuna.com