Access Control

Attribute-based Access Control

Prof.dr. Ferucio Laurențiu Țiplea

Fall 2023

Department of Computer Science
“Alexandru Ioan Cuza” University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Outline

- Introduction to ABAC
- Attributes and policies
- ABAC architectures
    - XACML
    - NGAC
- Conclusions

Introduction to ABAC

Criticisms of RBAC

- Defining a role can be quite a challenge: all the necessary permissions for a user who will take that role must be known a priori, as well as the role's position in the hierarchy;
- RBAC is static: RBAC assumes that, in most applications, permissions needed for an organization's roles change slowly over time, but users may enter, leave, and change roles rapidly. Using contextual information, such as time, location etc., is very difficult or even impossible;
- Deal with distributed applications: proper operation of RBAC requires that roles fall under a single administrative domain or have a consistent definition across multiple domains. So, distributed applications might be challenging;
- Role explosion: setting the role system for large organizations can be a complex process that can lead to "role explosion".

Criticisms of RBAC

- Temporary permissions: assigning individual permissions for short periods requires the creation of new roles;
- Permissions assigned only to roles: permissions can be assigned only to roles, not to objects and operations;
- Restrict access to data: RBAC can restrict access to certain actions of system but not to certain data.

RBAC is suitable for small or medium-sized organizations, with a well-understood role diagram, with clear assignments for each role. As the organization grows numerically and structurally, RBAC becomes inefficient.

What is ABAC

ABAC is a technology that has been used more or less sporadically by various researchers, mostly parallelizing RBAC (Kuhn et al. (2010); Karp et al. (2010)). There is currently no single accepted definition of ABAC!

A high-level description of ABAC by National Institute of Standards and Technology (Hu et al. (2019)):

> *Attribute Based Access Control: An access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, (optionally) environmental conditions, and a set of policies that are specified in terms of those attributes and conditions.*

Attributes and policies

Attributes

Attributes – characteristics of system entities.

- Subject attributes: identity, age, role, affiliation, security clearance etc.;
- Object attributes: characterize data and other resources, security classification;
- Environment attributes:
  - Are not properties of the subject or resources;
  - Are measurable characteristics that pertain to the operational or situational context in which access requests occur;
  - Are subject and object independent.

  Examples: current time, day of the week, threat level etc.

Subject and object attributes are common to all ABAC models!

Policies

According to Biswas et al. (2016), there are two main techniques for specifying authorization policies in ABAC:

- Logical formulas involving attribute values, such as in ABAC$_\alpha$, HGABAC, XACML. Example:

$$\begin{array}{lcl} \mathit{can\_access}(s, a, o) & \rightarrow & \mathit{role}(s) = \texttt{doctor} \\ & & \mathit{AND}\ \mathit{ward}(s) = \mathit{ward}(o) \\ & & \mathit{AND}\ (a = \texttt{read}\ \mathit{OR}\ a = \texttt{write}) \end{array}$$

- Enumeration, such as in Policy Machine (PM) and 2-sorted RBAC.

The enumeration technique may be more efficient when it comes to policy review!

ABAC architectures

ABAC architectures

ABAC implementations usually follow two general architectures to define a policy language and a processing model describing how to evaluate access requests (Ferraiolo et al. (2016)):

- eXtensible Access Control Markup Language (XACML);
- Next Generation Access Control (NGAC).

Both architectures encompass four layers of functional decomposition:

- enforcement;
- decision;
- access control data;
- administration.

eXtensible Access Control Markup Language

eXtensible Access Control Markup Language (XACML)

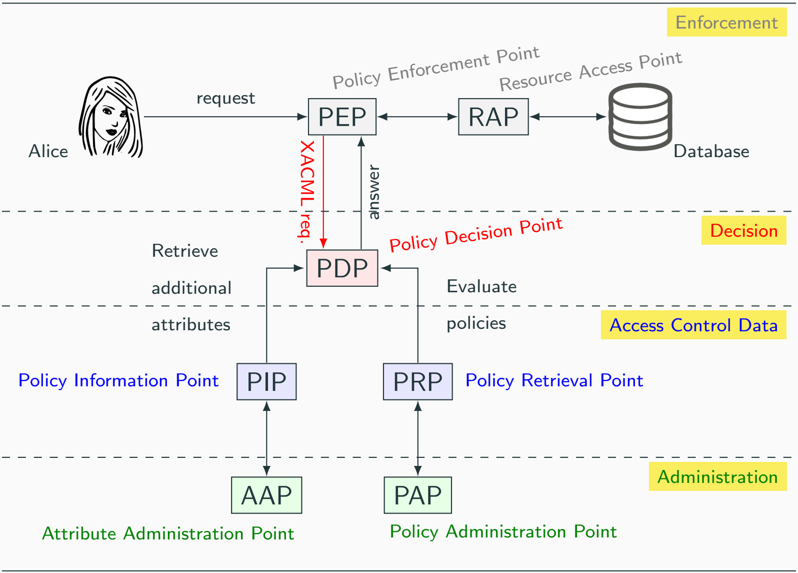
- Is a an ABAC policy language designed to express security policies and access requests to information ;
- The standard was published by OASIS and is now at version 3.0.

XACML can be used for:

- web services;
- digital rights management;
- enterprise security applications.

XACML does not handle user access, approval, or password management!

XACML architecture



XACML examples

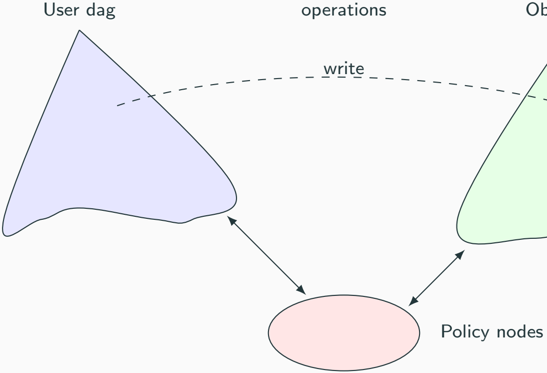
```
<Rule Effect="Permit" RuleId="rule1">
  <Condition>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <AttributeDesignator
             AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
             Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
             DataType="http://www.w3.org/2001/XMLSchema#string"
             MustBePresent="true"/>
      </Apply>
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
             Pamoda
      </AttributeValue>
    </Apply>
  </Condition>
</Rule>
```

XACML implementations

1. Sun XACML: partial XACML 2.0 implementation;
2. Heras AF: XACML 2.0 implementation;
3. Balana: partial XACML 3.0 implementation;
4. AuthzForce : fully XACML 3.0 compliant (all mandatory features);
5. Axiomatics Policy Server: XACML 2.0 and 3.0 implementation;
6. Oracle XACML: XACML 2.0 implementation;
7. IBM DataPower: XACML 2.0 implementation.

Next Generation Access Control

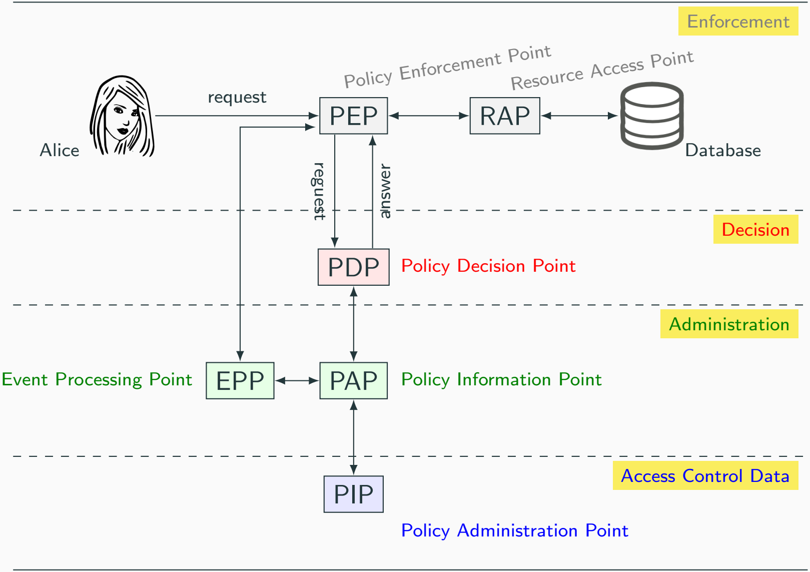
Next Generation Access Control (NGAC) uses directed acyclic graphs (dags) to represent the resources to protect, the different actors in the system, and how both worlds are tied together with permissions (INCITS (2020)).



NGAC features

- Can be configured to allow or disallow access based not only on object attributes, but also on environmental conditions such as time, location, and so on;
- Ability to set policies consistently to meet compliance requirements and the ability to set ephemeral policies;
- Can evaluate and combine multiple policies in a single access decision, while keeping its linear time complexity.

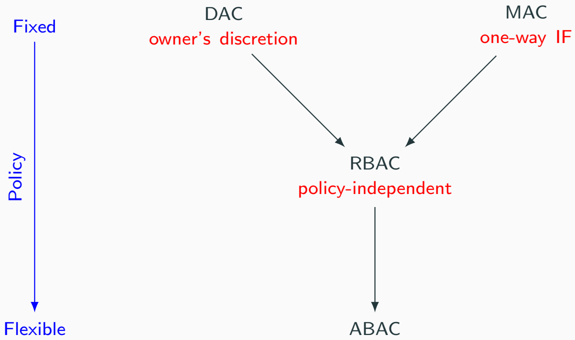
NGAC architecture



Conclusions

ABAC, RBAC, DAC, and MAC

ABAC can be configured to do RBAC: a role is just another attribute (details can be found in Jin et al. (2012)).



Concluding remarks

- From the multitude of access control technologies developed from the 1970s to the present, the technologies with significant practical relevance are: DAC, MAC, RBAC, ABAC;
- ABAC represents the latest milestone in the evolution of access control technologies;
- In practical applications, choosing the technology that covers the requirements minimally is recommended (if DAC or MAC is sufficient, it does not make sense to use RBAC or ABAC);
- The techniques presented so far should not be confused with decryption key distribution techniques to access the original data obtained by decryption.

References

Biswas, P., Sandhu, R., and Krishnan, R. (2016). Label-based access control: An abac model with enumerated authorization policy. In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, ABAC '16, page 1–12, New York, NY, USA. Association for Computing Machinery.

Ferraiolo, D., Chandramouli, R., Kuhn, R., and Hu, V. (2016). Extensible access control markup language (XACML) and next generation access control (NGAC). In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, ABAC '16, page 13–24, New York, NY, USA. Association for Computing Machinery.

Hu, V., Ferraiolo, D., Kuhn, D., Schnitzer, A., Sandlin, K., Miller, R., and Scarfone, K. (2019). Guide to attribute based access control (ABAC) definition and considerations. Technical report, National Institute of Standards and Technology Special Publication.

INCITS (2020). Next Generation Aaccess Control (NGAC). Technical Report 565-2020, American National Standards Institute.

Jin, X., Krishnan, R., and Sandhu, R. (2012). A unified attribute-based access control model covering DAC, MAC and RBAC. In Cuppens-Boulahia, N., Cuppens, F., and Garcia-Alfaro, J., editors, *Data and Applications Security and Privacy XXVI*, pages 41–55, Berlin, Heidelberg. Springer Berlin Heidelberg.

Karp, A., Haury, H., and Davis, M. (2010). From ABAC to ZBAC: The evolution of access control models. *Journal of Information Warfare*, 9(2):38–46.

Kuhn, D. R., Coyne, E. J., and Weil, T. R. (2010). Adding attributes to role-based access control. *Computer*, 43(6):79–81.