ABAC with Group Attributes and Attribute Hierarchies Utilizing the Policy Machine

2nd ACM Workshop on Attribute-Based Access Control (ABAC)
March 24, 2017

By
Smriti Bhatt, Farhan Patwa and Ravi Sandhu
Department of Computer Science
University of Texas at San Antonio
Outline

❖ Introduction
❖ Motivation
❖ Background & Related Work
❖ A Restricted HGABAC (rHGABAC) Model
❖ Policy Machine (PM) and its Architecture
❖ Authorization Architecture
❖ rHGABAC Utilizing Policy Machine
❖ Use Cases
❖ Policy Evaluation in PM
❖ Conclusion
Attribute-Based Access Control (ABAC)

- Access control based on attributes of users and objects
- Flexible and fine grained access control model
- Core Entities:
  - Users
  - Objects
  - Attributes (Users & Objects)
  - Permissions/Actions
  - Authorization Policy
Many different ABAC models

Authorization policy specification in ABAC Models

Additional components and capabilities in ABAC

User and Object Groups

Group Attributes and Group Hierarchy

Attribute Hierarchy

LAPs – Logical-Formula Authorization Policy

EAPs – Enumerated Authorization Policy

$c_{\text{ clearance}(u)} \geq c_{\text{ classification}(o)}$

$(u_{a_i}, o_{a_j})$
Motivation

❖ ABAC models and policies in real-world applications
❖ Enforcement of ABAC policies through existing ABAC frameworks and tools
  ❖ XACML
  ❖ Policy Machine
❖ Ease of policy and attribute administration and management
Background & Related Work

❖ HGABAC – A hierarchical attribute-based access control model
  ❖ User and Object Groups
  ❖ Group Attributes and Hierarchies
❖ HABE – A hierarchical attribute-based encryption mechanism
❖ LaBAC_H – Label-based access control model with hierarchy
  ❖ Hierarchical relationship among attribute values
Group Attributes and Hierarchies

- Group attributes
- Hierarchy among groups

![Diagram of user group hierarchy]

Fig 1. An Example of User Group Hierarchy Adapted from [*]

A partial ordering of *Range* of attribute values

Fig 2. An Example of Attribute Hierarchy

- a. User Attribute-value Hierarchy
- b. Object Attribute-value Hierarchy
Fig 3. rHGABAC Model Adapted from [1,2]

Fig 4. *rHGABAC* Model with Attribute Hierarchy
Unified attribute-based access control framework

Express and enforce variety of access control policies utilizing PM Policy

Configuration Points

- Commonly known and implemented access control policies (DAC, MAC, RBAC)
- Combinations of policies
- New access control policies

**PM Core Elements**
- Users
- Objects
- User Attributes
- Object Attributes
- Operations, Access Rights
- Processes
- Policy Classes

**PM Relations**
- Assignment
- Association
- Prohibition
- Obligation

✓ **assignment**—for specifying relationships between policies, users, and user attributes, objects and object attributes

✓ **association**—for defining policies through associations between user attributes and object attributes or objects through some operations
Fig 5. Architectural Components of PM Adapted from [*]

Fig 5. Architectural Components of PM Adapted from [*]

Fig 6. Authorization Architecture Utilizing PM and AE
Implementation

- PM Version 1.5
- Utilized PM Server (PAP + PDP) and PM Database (Active Directory)
- PM Agnostic Applications
- Need support for RESTful API in order to communicate to our Authorization Engine (AE)*
- Resources and their access points are abstracted within applications

rHGABAC Policy Configuration in PM

- User groups, user attributes and their values modeled as PM User Attributes
- Object groups, object attributes and their values modeled as PM Object Attributes
- Hierarchical relationships represented using PM’s assignment relation and containment property

Containment Property:

\[ x \text{ ASSIGN}^+ y \]

\[ x \text{ contained by } y \]
A simplified Policy Element Diagram in Policy Machine
Use Case – Group Attributes and Hierarchy

a. User Groups and their assigned user attributes and values

- Dev
  - skills = {Java}
  - depart = {Dev_Ops}

- Depl
  - skills = {C, C++, Java}
  - depart = {Dev_Ops}

b. Object Groups and their assigned object attributes and values

- Networking_Project
  - type = {General, Networking}

- Dev_Project
  - type = {General, Dev}

- Depl_Project
  - type = {General, Dev, Deploy}

© Smriti Bhatt

World-Leading Research with Real-World Impact!
Use Case – PM Graph

Fig 7. Group Hierarchy Policy Graph (Based on PM Graph Structure)
### Use Case – Policy

**Policy_{read}**

<table>
<thead>
<tr>
<th>User Attribute Values</th>
<th>Object Attribute Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT_Manager</td>
<td>Networking</td>
</tr>
<tr>
<td>IT</td>
<td>Networking</td>
</tr>
<tr>
<td>DevOps_Manager</td>
<td>Dev</td>
</tr>
<tr>
<td>Java</td>
<td>Dev</td>
</tr>
<tr>
<td>DevOps_Manager</td>
<td>Deploy</td>
</tr>
<tr>
<td>Java</td>
<td>Deploy</td>
</tr>
<tr>
<td>C</td>
<td>Deploy</td>
</tr>
<tr>
<td>C++</td>
<td>Deploy</td>
</tr>
<tr>
<td>CTO</td>
<td>General</td>
</tr>
</tbody>
</table>

**Authorization Policy Request and Response**

```
stack@pm-app1:/$ curl -s http://192.168.1.0:9000/echoGet -X GET -d
{"type":"hierarchical",
 "user":"user_IT2",
 "operation":"read",
 "object":"obj_Net1"
}
{"access":"granted"}
stack@pm-app1:/$
```
Use Case Extended with Attribute Hierarchy

a. Subgraph Showing Attribute Hierarchy in *skills* Attribute

b. Subgraph Showing Attribute Hierarchy in *type* Attribute
Use Case Extended with Attribute Hierarchy

<table>
<thead>
<tr>
<th>Policy_{read}</th>
<th>Policy_{read}</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Attribute Values</td>
<td>Object Attribute Values</td>
</tr>
<tr>
<td>IT_Manager</td>
<td>Networking</td>
</tr>
<tr>
<td>IT</td>
<td>Networking</td>
</tr>
<tr>
<td>DevOps_Manager</td>
<td>Dev</td>
</tr>
<tr>
<td>Java</td>
<td>Dev</td>
</tr>
<tr>
<td>C</td>
<td>Deploy</td>
</tr>
<tr>
<td>C++</td>
<td>Deploy</td>
</tr>
<tr>
<td>CTO</td>
<td>Deploy</td>
</tr>
<tr>
<td>General</td>
<td></td>
</tr>
</tbody>
</table>

Policy Without Attribute Hierarchy

<table>
<thead>
<tr>
<th>Policy_{read}</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Attribute Values</td>
</tr>
<tr>
<td>IT_Manager</td>
</tr>
<tr>
<td>IT</td>
</tr>
<tr>
<td>DevOps_Manager</td>
</tr>
<tr>
<td>Java</td>
</tr>
<tr>
<td>C++</td>
</tr>
<tr>
<td>CTO</td>
</tr>
<tr>
<td>General</td>
</tr>
</tbody>
</table>

Policy With Attribute Hierarchy
Policy Evaluation in PM

❖ Comparison of policy evaluation times for different ABAC policies in PM using our authorization architecture with AE

Average Policy Evaluation Time for ABAC Policies

<table>
<thead>
<tr>
<th>Policy</th>
<th>Avg. Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role-Centric ABAC</td>
<td>26.04</td>
</tr>
<tr>
<td>rHGABAC</td>
<td>27.04</td>
</tr>
<tr>
<td>rHGABAC with AH</td>
<td>26.57</td>
</tr>
</tbody>
</table>
Conclusion

❖ A restricted HGABAC model \((rHGABAC)\) presented and formalized as a single-value EAP

❖ Employed group attributes and group hierarchies, as well as attribute hierarchies in an ABAC model

❖ Presented a generalized authorization architecture for enforcement of ABAC policies

❖ \(rHGABAC\) simplifies Policy and Attribute management and administration in ABAC policies

❖ New versions of PM tool would provide better insights in new ways of expressing and enforcing ABAC policies
Thank you!!!
Questions???