MODALITY CONFLICT ANALYSIS IN XACML POLICY EVALUATION

Teo Poh Kuang1, Hamidah Ibrahim2, Fatimah Sidi3, and Nur Izura Udzi4

1Universiti Putra Malaysia, Malaysia, pohkuang1985@yahoo.com.my
2Universiti Putra Malaysia, Malaysia, hamidah.ibrahim@upm.edu.my
3Universiti Putra Malaysia, Malaysia, fatimah@upm.edu.my
4Universiti Putra Malaysia, Malaysia, izura@upm.edu.my

ABSTRACT. Modality conflict is one of the main issues in policy evaluation. Modality conflict arises when two or more policies that refer to the same subject, action, and resource but with modalities of opposite sign. Authorizations could be propagated according to the inheritance relationships between concepts not only based on subject, resource, and action, but also condition. Identifying the applicable policies and detecting the modality conflict when temporal and spatial constraints are specified in the policies have not received enough attention. Hence, in this paper an authorization propagation rule is proposed to identify the applicable policies during policy evaluation, which relies on inheritance relationships between concepts, on the basis of the partially ordered structures obtained by classifying subject, resource, action, and condition attributes. An effective authorization propagation rule can detect most of the modality conflicts that occur among the applicable policies.

Keywords: modality conflict, authorization propagation, inheritance

INTRODUCTION

With the increasing popularity of distributed systems and collaborative applications, there is a need for a conflict analysis method in policy evaluation. Traditional modality conflict is determined by authorizations of opposite effect (indicated by + and -) that is applied to the same subject, object, and action [Moffett, J. D., & Sloman, M. S., 1994; Lupu, E., & Sloman, M., 1997; Damianou, N. et al., 2002; Boutaba, R., & Aib, I., 2007; Damiani, E. et al., 2006]. Typically in a large distributed system, the authorization policies may be propagated according to the inheritance relationships between concepts which may cause inconsistencies.

Several works have been devoted to the topic of propagation of authorizations in distributed systems [Bertino, E. et al., 1998; Jajodia, S. et al., 2001; Damiani, E. et al., 2006; Adi, K. et al., 2009; Mohan, A. et al., 2011; Brodecki, B. et al., 2012; Shaikh, R. A. et al., 2016]. However, the concern of these works is only on the authorization propagation of the subject, resource, and action attributes, and not on the condition attribute. Adi, K. et al. (2009) argued that sometimes it is required to consider additional temporal as well as spatial constraints on the permission inheritance hierarchy in order to restrict policy permission. Hence, we propose an authorization propagation rule to identify the applicable policies during policy evaluation. The modality conflict is detected among explicit and implicit applicable policies. The authorization propagation rule relies on the inheritance relationships between concepts, on the basis
of the partially ordered structures obtained by classifying subject, resource, action, and condition attributes.

This paper is organized as follows. The related works are discussed in the following section. This is followed by the presentation of the modality conflict analysis process. The experiment result is presented in the next section. The last section concludes our work.

RELATED WORKS

Typically in a large distributed system, when a user sends a request to execute an action, if there is no explicit authorization specified for the user, there must be some way to propagate authorizations to the user [Jajodia et al., 2001]. In other words, the authorization policies may be propagated according to the inheritance relationships between concepts. Although the authorization propagation can derive the implicit policies, but it can result in unforeseen conflicts [Kamoda, H. et al., 2005]. Hence, it is necessary to detect and resolve the modality conflict when both a denial and a permission are specified among the explicit and implicit policies.

Past works [Proctor, S., 2004; Priebe, T. et al., 2007; Dong, C. et al., 2008; Liu, A. X. et al., 2011; Fatema, K., & Chadwick, D., 2014; Ammar, N. et al., 2015; Ngo, C. et al., 2015 supported traditional modality conflict detection which has no hierarchical structure for organizing subject, action, resource, and condition. Therefore, all the policies must be defined instance by instance, which will be burdensome in large systems since no authorization can be propagated according to the inheritance relationships. A number of works have been devoted to the topic of propagation of authorizations based on the inheritance relationships between concepts [Bertino, E. et al., 1998; Jajodia, S. et al., 2001; Damiani, E. et al., 2006; Adi, K. et al., 2009; Mohan, A. et al., 2011; Brodecki, B. et al., 2012; Shaikh, R. A. et al., 2016].

Bertino, E. et al. (1998) argued that the lower level of role is its lower position within an organization, it is reasonable to assume that the access permissions given to a role subsumed the access permissions given to all roles with a lower position in the hierarchy. The same concept is applied to object hierarchy. Jajodia, S. et al. (2001) presented a unified framework which allows the specification of both positive and negative authorizations and incorporate notions of authorization derivation, conflict resolution, and authorization decision strategies by exploiting the hierarchical structures of attributes (roles, user group, and resources). Damiani, E. et al. (2006) exemplified modality conflict arising from "part-of" relations. The modality conflict identified by these authors can be resolved by considering the concept hierarchy used for propagating authorizations. The specificity principle being applied is based on the notion of domain nesting principle whenever this relationship exists. Mohan, A. et al. (2011) applied descending propagation by evaluating the child nodes which have authorization decision different from the requested resource node. Brodecki, B. et al. (2012) proposed an algorithm to detect modality conflict between two policies with opposite authorization decisions when descending propagation is applied with the knowledge of the hierarchy of subjects and resources. Shaikh, R. A. et al. (2016) proposed a novel method which aimed at detecting modality conflict which occurs among the access control policies when the concept of role hierarchy and permission inheritance are introduced in the access control model. These works [Bertino, E. et al., 1998; Jajodia, S. et al., 2001; Damiani, E. et al., 2006; Adi, K. et al., 2009; Mohan, A. et al., 2011; Brodecki, B. et al., 2012; Shaikh, R. A. et al., 2016] analyzed authorization propagation policy according to the subsumption relationships between concepts on the basis of the partially ordered structures obtained based on subject hierarchy, action hierarchy or resource hierarchy only and are limited to simple condition evaluation in which string equal function is used.
Adi, K. *et al.* (2009) argued that sometimes it is required to consider additional temporal as well as spatial constraints on the permission inheritance hierarchy in order to restrict policy permission. In addition, complex condition elements such as semantic relationships between spatial element i.e. the requestor’s location information, or temporal element i.e. the requestor’s access time are necessary to take into account in the modality conflict detection process.

**THE MODALITY CONFLICT PROCESS**

Figure 1 shows the overall general process flow of our proposed modality conflict model that aims to identify the explicit and implicit policies based on a given request. The modality conflict is detected among the applicable explicit and implicit policies during execution of a request. When a user sends a request to access the resources of an organization, the authorization module will determine which policy is applicable to the particular request. These applicable policies can have a different or even conflicting authorization decision for the request.

XACML policy language supports policy or rule combination algorithms, which evaluate the applicable policy based on the logic of the algorithm. Authorizations are automatically propagated along subject, resource, action, and location hierarchies and the authorization flows are always from the parent towards its child nodes. Hence, the implicit policies can be derived based on these hierarchies. Our work provides an authorization propagation rule in order to investigate the class-subclass relationships of a subject, resource, action and location of a request and a policy before the applicable policy is identified. The proposed authorization propagation rule used in our model is as follow:

\[
\text{Subject}_\text{req} \leq \text{Subject}_{\text{pol}} \land \text{Resource}_\text{req} \leq \text{Resource}_{\text{pol}} \land \text{Action}_\text{req} \leq \text{Action}_{\text{pol}} \land \text{Location}_\text{req} \leq \text{Location}_{\text{pol}}
\]

where \( Subject_{req} \), \( Resource_{req} \), \( Action_{req} \), and \( Location_{req} \) are the subject, resource, action and location of the request respectively; while \( Subject_{pol} \), \( Resource_{pol} \), \( Action_{pol} \), and \( Location_{pol} \) are the subject, resource, action and location of the policy respectively. The operator \( \leq \) represents the subsumption operator.

Our modality conflict detection algorithm is shown in Figure 2. The proposed authorization propagation rule stated that the explicit and implicit policies will be retrieved if its conditions are obeyed. The conflict resolution is needed in order to resolve the modality conflict before an authorization decision is returned. XACML defines four types of predefined combining algorithm to automatically resolve modality conflict: "Permit-Overrides", "Deny-
Overrides", "First-Applicable", and "Only-One-Applicable". In the following an example is given to illustrate how modality conflict exists among applicable policies when authorization is being propagated. Table 1 presents three explicit access control policies and a request is presented in Table 2.

### Figure 2. The Modality Conflict Detection Algorithm.

**Table 1: The University Policies.**

<table>
<thead>
<tr>
<th>Policy No.</th>
<th>Effect</th>
<th>Subject</th>
<th>Resource</th>
<th>Action</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Permit</td>
<td>RA</td>
<td>ExternalGrades</td>
<td>Assign</td>
<td>(Location = Association) &amp; (Time ≤ 12P.M. &amp; Time ≤ 2P.M.) &amp; (Email = <a href="mailto:p23452@upm.edu.my">p23452@upm.edu.my</a>)</td>
</tr>
<tr>
<td>P2</td>
<td>Permit</td>
<td>AssociateProfessor</td>
<td>Grades</td>
<td>Assign &amp; View</td>
<td>(Location = GraduateSchool) &amp; (Time ≥ 12P.M. &amp; Time ≤ 1P.M.)</td>
</tr>
<tr>
<td>P3</td>
<td>Deny</td>
<td>FacultyMember</td>
<td>Grades</td>
<td>Assign &amp; View</td>
<td>(Location = School) &amp; (Time ≥ 12P.M. &amp; Time ≤ 1P.M.)</td>
</tr>
</tbody>
</table>

**Table 2: The Request for Policy Evaluation.**

<table>
<thead>
<tr>
<th>Request No.</th>
<th>Subject</th>
<th>Resource</th>
<th>Action</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req1</td>
<td>AssociateProfessor</td>
<td>InternalGrades</td>
<td>Assign</td>
<td>(Location = GraduateSchool) &amp; (Time ≥ 12P.M. &amp; Time ≤ 1P.M.)</td>
</tr>
</tbody>
</table>

Based on Req1, the proposed solution retrieved two explicit policies, which are P2 and P3. Table 3 presents the implicit policies which are derived from P2 and P3. P2_{implicit} is a policy derived based on P2. All the elements of Req1 matched exactly with the elements of P2 except for InternalGrades which is a subclass of Grades in P2. P3_{implicit} is a policy derived based on P3. Faculty_Member in P3 is a superclass of AssociateProfessor in Req1 since AssociateProfessor is a child node of Faculty_Member. InternalGrades in Req1 is a subclass of Grades in P3. Assign in Req1 is matched to Assign in P3. While GraduateSchool in Req1 is a subclass of School in P3. The proposed solution has identified P2_{implicit} and P3_{implicit} as the applicable policies. However, the effect of P2_{implicit} conflicts with the effect of P3_{implicit}. In this case, the policy combining algorithm, "Deny-Overrides" is chosen to resolve the modality conflict. Thus, "Deny" is returned as the authorization decision for Req1.

**Table 3: The Implicit Policies Derived from P2 and P3 based on Req1.**

<table>
<thead>
<tr>
<th>Policy No.</th>
<th>Effect</th>
<th>Subject</th>
<th>Resource</th>
<th>Action</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2_{implicit}</td>
<td>Permit</td>
<td>AssociateProfessor</td>
<td>InternalGrades</td>
<td>Assign</td>
<td>(Location = GraduateSchool) &amp; (Time ≥ 12P.M. &amp; Time ≤ 1P.M.)</td>
</tr>
<tr>
<td>P3_{implicit}</td>
<td>Deny</td>
<td>AssociateProfessor</td>
<td>InternalGrades</td>
<td>Assign</td>
<td>(Location = GraduateSchool) &amp; (Time ≥ 12P.M. &amp; Time ≤ 1P.M.)</td>
</tr>
</tbody>
</table>
EXPERIMENT RESULT

We have performed a simple analysis to evaluate the accuracy of the proposed authorization rule. Table 4 highlights the performance of our proposed solution and the Sun's XACML implementation [Proctor, S., 2004] with respect to the results of \( P, R, \) and \( F \) in detecting modality conflict among the applicable policies. The \textit{CodeA} which is taken from Liu [Liu, A. X. et al., 2011] and the \textit{UniversityStoller} for the university which is taken from Stoller [Stoller, S. D. et al., 2007] are modified by adding additional deny rule for each policy for modality conflict detection. The experiment results are analyzed at varying value of similarity thresholds, \( \tau \). A higher value of \( \tau \) implies stricter matching requirements between the string elements of a request and a policy. Sun's XACML implementation achieved 0% \( P, 0\% R, \) and 0% \( F \) in detecting the modality conflict for \textit{CodeA}. Even for \textit{UniversityStoller} policy, the Sun's XACML implementation achieved lower percentage of \( P, R, \) and \( F \) in detecting the modality conflict compared to the proposed solution. Sun's XACML implementation supports simple string equal matching function and does not consider the inheritance relationships which can be identified through subject, resource, action, and location hierarchy. Therefore, the applicable policies are not retrieved and this affected the modality conflict detection. Our proposed solution achieved higher percentage of \( P, R, \) and \( F \) compared to the Sun's XACML implementation in detecting modality conflict even when the similarity threshold is set to a higher value. Our proposed solution enables the authorizations to be propagated from a node to all its descendants which are semantically related to it according to the inheritance relationships which are identified not only through subject, resource, action hierarchy, but also condition (i.e. location). Therefore, our proposed solution can identify all possible explicit and implicit applicable policies for a request. This indicates that our proposed solution is better compared to the Sun's XACML implementation.

Table 4: Precision, \( (P) \), Recall, \( (R) \), and F-measure, \( (F) \) for Modality Conflict Detection by the Proposed Solution and Sun’s XACML Implementation.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Evaluation Metric</th>
<th>Proposed Solution</th>
<th>Sun’s XACML Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision ( (P) )</td>
<td>( 0.2 ) 0.4 0.6 0.8 1.0</td>
<td>( 0 )</td>
</tr>
<tr>
<td>CodeA</td>
<td>60.80</td>
<td>75.00</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>72.27</td>
<td>72.27</td>
<td>72.27</td>
</tr>
<tr>
<td></td>
<td>FMeasure ( (F) )</td>
<td>37.90</td>
<td>46.00</td>
</tr>
<tr>
<td>UniversityStoller</td>
<td>72.25</td>
<td>100 100 100 100</td>
<td>( 0 )</td>
</tr>
<tr>
<td></td>
<td>67.65</td>
<td>57.99</td>
<td>57.99</td>
</tr>
<tr>
<td></td>
<td>FMeasure ( (F) )</td>
<td>76.09</td>
<td>73.41</td>
</tr>
</tbody>
</table>

CONCLUSION

This paper addresses a model and an algorithm that support the authorization propagation rule which explores the inheritance relationships of a subject, resource, action, and condition to retrieve the explicit and implicit applicable policies for a given request. Because of the complexity of semantic policy repositories and the variation of user access privileges, modality conflict may arise in a group of access control policies. The proposed solution can be further enhanced by considering other factors which could affect the authorization decisions such as obligations for which some actions should be launched once certain conditions are satisfied.

REFERENCES


