How to cite this paper:

Teo Poh Kuang, Hamidah Ibrahim, Fatimah Sidi, & Nur Izura Udzi. (2017). Modality conflict analysis in XACML policy evaluation in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference on Computing & Informatics (pp 708-713). Sintok: School of Computing.

MODALITY CONFLICT ANALYSIS IN XACML POLICY EVALUATION

Teo Poh Kuang[1], Hamidah Ibrahim[2], Fatimah Sidi[3], and Nur Izura Udzi[4]

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

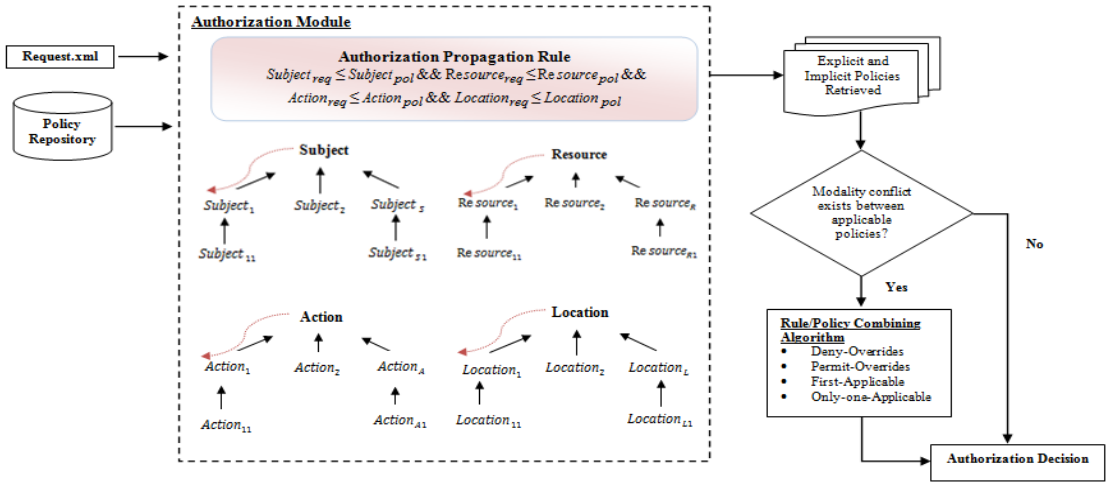

Figure 1. The Modality Conflict Detection Model.

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:

$$Subject_{req} \leq Subject_{pol} \;\&\&\; Resource_{req} \leq Resource_{pol} \;\&\&\; Action_{req} \leq Action_{pol} \;\&\&\; Location_{req} \leq Location_{pol} \quad (1)$$

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.

Input:
Req: A request from a user in *OrgA*
Pi: A set of policies, {*P*1, *P*2, …, *Pm*} in *OrgB*
Hierarchy: Subject hierarchy, *Hsub*, resource hierarchy, *Hres*, action hierarchy, *Hact*, location hierarchy, *Hloc*
Output: *Conflict*/*Not Conflict*
For each attribute value of *Req* ($Subject_{req}$, $Resource_{req}$, $Action_{req}$, $Location_{req}$) and *Pi* ($Subject_{pol}$, $Resource_{pol}$, $Action_{pol}$, $Location_{pol}$)
  Apply *Hsub*, *Hres*, *Hact*, and *Hloc*
  If ($Subject_{req} \leq Subject_{pol}$) && ($Resource_{req} \leq Resource_{pol}$) && ($Action_{req} \leq Action_{pol}$)
    If $Location_{req}$!= null && $Location_{pol}$!= null
      If ($Location_{req} \leq Location_{pol}$)
        Then retrieve *Pi* and store it into an array, *Array*
      Else check next policy
    ElseIf $Location_{pol}$ = null
      Then retrieve *Pi* and store it into an array, *Array*
    Else check next policy
  Else check next policy
End For
For each applicable policy of *Array*, compare the effect of the applicable policy and check whether modality conflict exists
  If *Effect_Pi* ≠ *Effect_Pj* in policy set level
    Return *Conflict*
    Resolve modality conflict based on the policy combining algorithm as stated in the policy
  ElseIf *Effect_Pi* ≠ *Effect_Pj* in policy level
    Return *Conflict*
    Resolve modality conflict based on the rule combining algorithm as stated in the rule
  Else Return *No Conflict*
End For

Figure 2. The Modality Conflict Detection Algorithm.

Table 1: The University Policies.

Policy No.	Effect	Subject	Resource	Action	Condition
*P*1	Permit	RA	ExternalGrades	Assign ∨ View	(Location = Association) ∧ (Time≥ 12P.M. ∧ Time ≤ 2P.M.) ∧ (Email = gs23442@upm.edu.my)
*P*2	Permit	AssociateProfessor	Grades	Assign ∨ View	(Location = GraduateSchool) ∧ (Time≥ 12P.M. ∧ Time ≤ 1P.M.)
*P*3	Deny	Faculty_Member	Grades	Assign ∨ View	(Location = School) ∧ (Time≥ 12P.M. ∧ Time ≤ 1P.M.)

Table 2: The Request for Policy Evaluation.

Request No.	Subject	Resource	Action	Condition
*Req*1	AssociateProfessor	InternalGrades	Assign	(Location = GraduateSchool) ∧ (Time =12.30P.M.)

Based on *Req*1, the proposed solution retrieved two explicit policies, which are *P*2 and *P*3. Table 3 presents the implicit policies which are derived from *P*2 and *P*3. $P2_{implicit}$ is a policy derived based on *P*2. All the elements of *Req*1 matched exactly with the elements of *P*2 except for *InternalGrades* which is a subclass of *Grades* in *P*2. $P3_{implicit}$ is a policy derived based on *P*3. *Faculty_Member* in *P*3 is a superclass of *AssociateProfessor* in *Req*1 since *AssociateProfessor* is a child node of *Faculty_Member*. *InternalGrades* in *Req*1 is a subclass of *Grades* in *P*3. *Assign* in *Req*1 is matched to *Assign* in *P*3. While *GraduateSchool* in *Req*1 is a subclass of *School* in *P*3. 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 *Req*1.

Table 3: The Implicit Policies Derived from *P*2 and *P*3 based on *Req*1.

Policy No.	Effect	Subject	Resource	Action	Condition
$P2_{implicit}$	Permit	AssociateProfessor	InternalGrades	Assign	(Location=GraduateSchool) ∧ (Time≥ 12P.M. ∧ Time ≤ 1P.M.)
$P3_{implicit}$	Deny	AssociateProfessor	InternalGrades	Assign	(Location= GraduateSchool) ∧ (Time≥ 12P.M. ∧ Time ≤ 1P.M.)

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 *CodeA* which is taken from Liu [Liu, A. X. *et al*., 2011] and the *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, τ. A higher value of τ 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 *CodeA*. Even for *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.

Policy	Evaluation Metric	Percentage (%)					
		Proposed Solution					Sun's XACML Implementation
		Similarity Threshold (τ)					
		0.2	0.4	0.6	0.8	1.0	
CodeA	Precision (*P*)	60.00	75.00	100	100	100	0
	Recall (*R*)	27.27	27.27	27.27	27.27	27.27	0
	F-Measure (*F*)	37.50	40.00	42.86	42.86	42.86	0
UniversityStoller	Precision (*P*)	72.73	100	100	100	100	0
	Recall (*R*)	67.05	57.99	57.99	57.99	57.99	7.60
	F-Measure (*F*)	76.09	73.41	73.41	73.41	73.41	14.12

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

Adi, K., Bouzida, Y., Hattak, I., Logrippo, L., & Mankovskii, S. (2009). Typing for Conflict Detection in Access Control Policies. *Proceedings of the 4th International MCeTECH Conference on eTechnologies* (*MCeTECH*), 212-226.

Ammar, N., Malik, Z., Bertino, E., & Rezgui, A. (2015). XACML Policy Evaluation with Dynamic Context Handling. *Journal of IEEE Transactions on Knowledge and Data Engineering*, 27(9), 2575-2588.

Bertino, E., Buccafurri, F., Ferrari, E., & Rullo, P. (1998). An Authorization Model and its Formal Semantics. *Proceedings of the 5th European Symposium on Research in Computer Security* (*ESORICS*), 127-142.

Boutaba, R. & Aib, I. (2007). Policy Based Management: A Historical Perspective. *Journal of Network and Systems Management*, 15(4), 447-480.

Brodecki, B., Szychowiak, M., & Sasak, P. (2012). Security Policy Conflicts in Service Oriented Systems. *Journal of New Generation Computing*, 30(2-3), 215-240.

Damiani, E., di Vimercati, S. D. C., Fugazza, C., & Samarati, P. (2006). Modality Conflicts in Semantics Aware Access Control. *Proceedings of the 6th International Conference on Web Engineering* (*ICWE*), 249-256.

Damianou, N., Bandara, A., Sloman, M., & Lupu, E. (2002). A Survey of Policy Specification Approaches. *Technical Report, Department of Computing, Imperial College of Science Technology and Medicine, London*.

Dong, C., Russello, G., & Dulay, N. (2008). Flexible Resolution of Authorisation Conflicts in Distributed Systems. *Proceedings of the 19th International Workshop on Distributed Systems: Operations and Management* (*DSOM*), 95-108.

Fatema, K. & Chadwick, D. (2014). Resolving Policy Conflicts-Integrating Policies from Multiple Authors. *Proceedings of the International Conference on Advanced Information Systems Engineering* (*CAiSE*), 310-321.

Jajodia, S., Samarati, P., Sapino, M. L., & Subrahmanian, V. (2001). Flexible Support for Multiple Access Control Policies. *Journal of ACM Transactions on Database Systems* (*TODS*), 26(2), 214-260.

Kamoda, H., Yamaoka, M., Matsuda, S., Broda, K., & Sloman, M. (2005). Policy Conflict Analysis using Free Variable Tableaux for Access Control in Web Services Environments. *Proceedings of the Policy Management for the Web Workshop at the 14th International World Wide Web Conference* (*WWW*), 121-126.

Liu, A. X., Chen, F., Hwang, J., & Xie, T. (2011). Designing Fast and Scalable XACML Policy Evaluation Engines. *Journal of IEEE Transactions on Computers*, 60(12), 1802-1817.

Lupu, E. & Sloman, M. (1997). Conflict Analysis for Management Policies. *Proceedings of the 5th IFIP/IEEE International Symposium on Integrated Network Management*, 430-443.

Moffett, J. D., & Sloman, M. S. (1994). Policy Conflict Analysis in Distributed System Management. *Journal of Organizational Computing and Electronic Commerce*, 4(1), 1-22.

Mohan, A., Blough, D. M., Kurc, T., Post, A., & Saltz, J. (2011). Detection of Conflicts and Inconsistencies in Taxonomy Based Authorization Policies. *Proceedings of the 2011 IEEE International Conference on Bioinformatics and Biomedicine* (*BIBM*), 590-594.

Ngo, C., Demchenko, Y., & Laat, C. D. (2015). Decision Diagrams for XACML Policy Evaluation and Management. *Journal of Computers and Security*, 49, 1-16.

Priebe, T., Dobmeier, W., Schläger, C., & Kamprath, N. (2007). Supporting Attribute Based Access Control in Authorization and Authentication Infrastructures with Ontologies. *Journal of Software*, 2(1), 27-38.

Proctor, S. (2004). Sun's XACML Implementation. URL: *http://sunxacml.sourceforge.net/*.

Shaikh, R. A., Adi, K., & Logrippo, L. (2016). A Data Classification Method for Inconsistency and Incompleteness Detection in Access Control Policy Sets. *International Journal of Information Security*, 1-23.

Stoller, S. D., Yang, P., Ramakrishnan, C. R., & Gofman, M. I. (2007). Efficient Policy Analysis for Administrative Role Based Access Control. *Proceedings of the 14th ACM Conference on Computer and Communications Security* (*CCS*), 445-455.