Ontology-based Evaluation of ABAC Policies for Inter-Organizational Resource Sharing

Tushar Gupta
Indian Institute of Technology Kharagpur
Kharagpur, India
gupta.tushar@iitkgp.ac.in

Shamik Sural
Indian Institute of Technology Kharagpur
Kharagpur, India
shamik@cse.iitkgp.ac.in

ABSTRACT

Attribute-based Access Control (ABAC), as the name suggests, determines whether an access request be granted based on the attributes or characteristics of the requesting user, that of the requested resource, and the environmental condition in which the request is generated. An important advantage of such an identity-agnostic model is that access control can be imposed even on users from other organizations if they are able to prove their attributes to the reference monitor of the organization whose resources are being accessed. It would, however, require a mechanism for mapping the attributes and their values between these two organizations. We propose an ontology based method for addressing this requirement. Besides meeting the needs of collaborative accesses, we show that such an approach also naturally supports enforcement of hierarchical ABAC policies as well as flexibility in policy enforcement. Both the problems of inter-organizational access control and hierarchical ABAC have so far attracted limited attention in the literature.

CCS CONCEPTS

• **Security and privacy → Access control**; **Software and application security**.

KEYWORDS

Attribute-based Access Control, Ontology, Attribute Hierarchy, Policy Relaxation, Digital Signature

ACM Reference Format:
Tushar Gupta and Shamik Sural. 2022. Ontology-based Evaluation of ABAC Policies for Inter-Organizational Resource Sharing. In *Proceedings of ACM Conference on Computer and Communications Security (ACM CCS).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Attribute-based access control (ABAC) is receiving increasing attention in recent years especially in large organizations due to its flexibility and ability to express complex authorization rules with relative ease. ABAC is based on attributes or characteristics of the users and resources (interchangeably called objects) in the system as well as those of the environment [1]. Any ABAC policy requires specific combinations of the values of user, resource and environment attributes to authorize an access request. Owing to this property, ABAC systems are amenable to easy sharing of resources among multiple organizations through presentation of verifiable attributes.

A major challenge, however, is that the notions and terminologies used for describing these attributes vary widely across organizations. As a result, it becomes difficult to allow inter-organization resource sharing even if the organization managing a certain set of resources (let's call it the host organization) is willing to share them with others (call these as guest organizations). While the ABAC policy is written using the terminology of the host organization, users from a guest organization may have different attributes even if they are semantically close. It may be noted that, maintaining guest organization specific policies does not scale and is also a system administration nightmare for the host.

In this paper, we address the above-mentioned problem by using a domain-specific ontology to capture the meanings of the attributes and depicting the relationships among them in a structured manner, thereby enabling appropriate evaluation of ABAC policies when an access request is originated from a guest organization user. We also use the relationships defined in the ontology to specify a hierarchy among the attributes in an organization as well as a distance between their various levels. Thus, besides facilitating inter-organization collaborative resource sharing, we show that the proposed approach provides a means for supporting hierarchical ABAC and also enables controlled relaxation in rule matching while enforcing an ABAC policy.

2 ACCESS REQUEST HANDLING

In Figures 1 and 2, we show sequence diagrams for how access requests from users internal to the host organization and those of guest organizations are handled. Since the host organization has all the attribute information about its internal users, these are simply checked against the ABAC rules and a decision can be taken whether to grant or deny access. On the other hand, the external users need to present their attribute values digitally signed by their respective organizations along with the access request (inter-organizational trust is established through prior verification of public keys). The same is first verified. If successful, an access decision is taken based on these attribute values and the prevailing ABAC rules similar to the case for internal users.

It can be observed from the figures that the same ABAC policy is used seamlessly to handle access requests from internal as well as external users once the attributes of the external users are verified. However, for the external users, the step marked as *Evaluate*

ACM CCS, Nov 07–11, 2022, Los Angeles, CA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

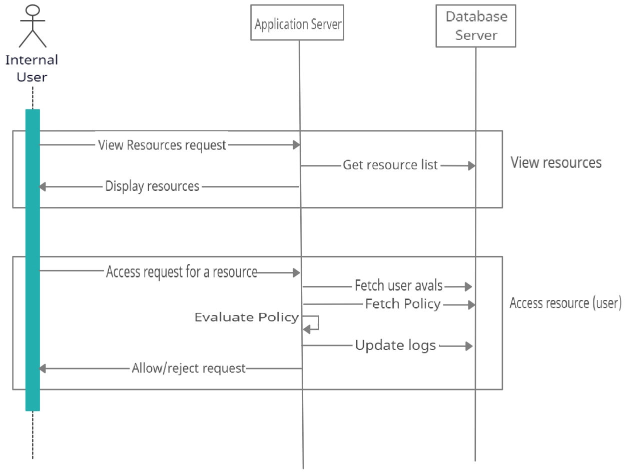

Figure 1: Handling access requests from internal users

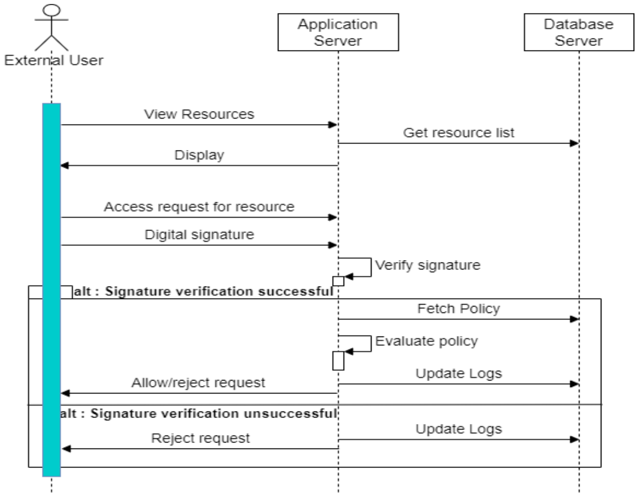

Figure 2: Handling access requests from external users

Policy in Figure 2 will require to map the attributes of the guest organization to those of the host organization. Even for internal users, there could be a hierarchy of attributes. We next explain how an ontology can be used to handle both these requirements.

3 ONTOLOGY AND ATTRIBUTE HIERARCHY

In our proposed approach, the host organization maintains an ontology for linking its own attribute names and their values with those of the participating guest organizations. With new guest organizations participating in resource sharing, such an ontology can be further enhanced without affecting the existing participants. We explain the process using an illustrative ontology for educational organizations.

We consider ontologies covering two major aspects of typical educational institutions: the academic divisions (Figure 3) and the roles or designations of the users associated with such organizations (Figure 4). We include varying terminologies from different institutions to make the ontology more realistic. The ontologies capture the relationship between the corresponding attributes.

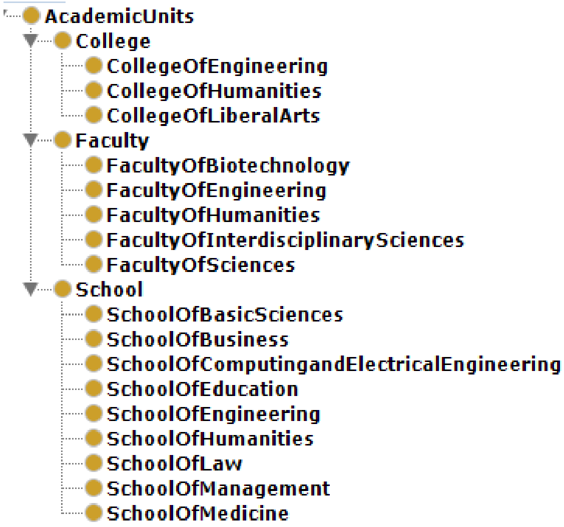

Figure 3: Illustrative ontology for academic units

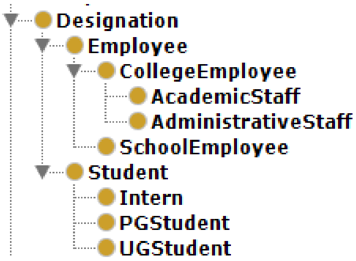

Figure 4: Illustrative ontology for user designations

In the figures, the leaf nodes can have various possible values for the attributes, e.g., *professor* or *director* for the attribute *designation*. Similarly, *ME*, *ECE* and other departments are some of the possible values for the academic unit *School of Engineering*. Note that, similar academic units are referred to differently in different institutions or universities such as School of Engineering or College of Engineering or Faculty of Engineering. They represent generic academic disciplines and contain individual *departments* under them devoted to specific disciplines.

The graph structure of the ontology allows us to define a hierarchy among the attributes and their values. Each element of the ontology is either an instance of a class, or a subclass derived directly or indirectly from a topmost (root) class. The level of an element in the hierarchy denotes the distance of the element node from the root. The distance between two nodes is the minimum number of steps required to reach one node from the other, moving to a subclass, superclass or instance (if the current node is a class), or parent class (if the current node is an instance) at each step. A node X is said to be at a higher level in the hierarchy than a node Y if the distance of X from the root is less than the distance of Y from the root.

A user U_1 is said to be at a higher level than a user U_2 with respect to an attribute A if U_1 has a value V_1 for A, U_2 has a value V_2 for A, and V_1 is at a higher level than V_2 in the hierarchy. Similarly, an ABAC rule P_1 is said to be at a higher level than a rule P_2 with respect to a (user, resource or environmental) attribute A if P_1 has a value V_1 for A, P_2 has a value V_2 for A, and V_1 is at a higher level than V_2 in the hierarchy with other attributes of P_1 either at the same or higher level than those of P_2. For example, a user associated with the *College of Engineering* can be said to be at a higher level with respect to the attribute *department* than a user associated with *Electrical Engineering*, which is part of the *College of Engineering*. It is easy to see that the notion of *higher level* imposes a partial order on the possible attribute values.

Hierarchical ABAC can thus be supported (for both internal as well as external users) with the help of the ontology and the above-mentioned definition of levels. A user attribute condition $A_u = V_u$ of an ABAC rule is satisfied by a user U if the user has a value $V_u^{'}$ for the attribute A_u such that $V_u^{'}$ is at a higher or the same level in the hierarchy as V_u. Likewise, a user condition $A_u = V_u$ of a rule is satisfied by a user U if the user has a value $V_u^{'}$ for the attribute A_u such that V_u is at a higher or same level in the hierarchy than $V_u^{'}$.

The distance between two attribute values in the ontology can also be used to define a measure of (dis)similarity between them. For a given value V, a value V_{d1} is more closely related to V than a value V_{d2} when $d1 \leq d2$. We may choose a maximum allowed distance D and get all attribute values at a distance $d \leq D$ from V in a list L_V^D. The values in L_V^D are the possible values that V can be mapped to within a relaxation distance D. Similarly, for an attribute A, we get the list of attribute mappings L_A^D by querying the ontology. These mappings are then used for evaluating the ABAC policy as follows. Given a maximum allowed distance D, a user condition $< A_P^u, V_P^u >$ of a rule P is satisfied if A_P^u is contained in the attribute mapping list $L_{A^u}^D$ and V_P^u is contained in the value mapping list $L_{V^u}^D$ for any user attribute-value pair $< A^u, V^u >$.

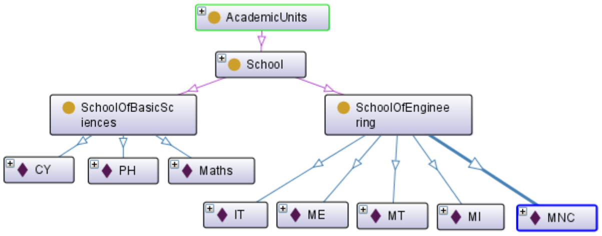

Figure 5: Attribute hierarchy example

Since the resources are owned by the host organization and the environmental context is independent of any particular organizations, we use the resource and environment attribute-value pairs that are defined by the host organization itself and an ontology is not required for them.

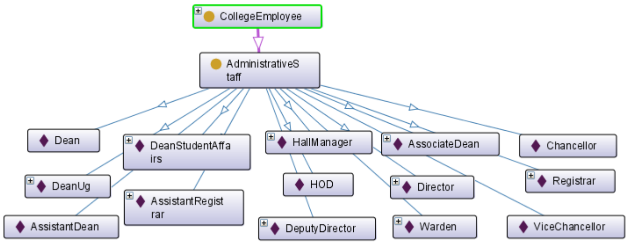

Figure 6: Designation hierarchy example

4 USE CASE AND RESULTS

The relevant portion of the ontology involving the values used in an example use case are shown in Figures 5 and 6. The resource here is named as *mechanics.pdf* and its attribute-value pair is $Department = ME$. The ABAC policy P consists of three rules P1, P2, and P3, all of which have a single resource condition: $Department = ME$, but the user conditions and operations allowed differ as follows:

- P1: $Department = ME \wedge Operation = write$
- P2: $Department = School \wedge Operation = read$
- P3: $Designation = AssistantDean \wedge Department = SchoolOfEngineering \wedge Operation = append$

Let there be a user U1 with attribute-value pairs as follows: $Designation = HOD \wedge Department = SchoolOfBasicSciences$. The various accesses can be evaluated as follows:

- P1: Not satisfied using the attribute hierarchy.
- P2: From Figure 5, it can be seen that the value *School* in user condition $Department = School$ of P2 is an ancestor of the user value $SchoolOfBasicSciences$ for attribute $Department$ of U1 in the hierarchy. So, P2 is satisfied and the operation *read* is allowed for the user.
- P3: Not satisfied using only the attribute hierarchy. However, from Figure 6, it can be seen that the value $AssistantDean$ in user condition $Designation = AssistantDean$ of P3 is at a distance of 2 from the user value HOD for attribute $Designation$ of U1. Also, in Figure 5, we find that the user condition $Department = SchoolOfEngineering$ of P3 is also at a distance of 2 from the user value $SchoolOfBasicSciences$ for attribute $Department$ of U1. So, if a relaxation of distance = 2 is allowed, the rule P3 will also be satisfied by user U1.

We implemented the proposed approach in Python and executed on an Intel Core i5 machine having 8GB RAM. The average execution time was between 0.5 to 0.9 milliseconds for external users depending on the value of the relaxation distance. For internal users, since digital signature verification is not necessary, the required time was less by a factor of 50%.

5 FUTURE WORK

The approach can be further enhanced by creating an ontology at scale, with multiple properties that can effectively capture the complex relationships among various attributes. Fully automated adjustment of relaxation distance based on the sensitivity level of objects and trustworthiness of users is another interesting research direction.

REFERENCES

[1] Daniel Servos and Sylvia L. Osborn. 2017. Current Research and Open Problems in Attribute-Based Access Control. *Comput. Surveys* 49, 4 (2017), 65:1–65:45. https://doi.org/10.1145/3007204