Towards Attribute-based Access Control Policy Engineering Using Risk*

Leanid Krautsevich, Aliaksandr Lazouski, Fabio Martinelli, and Artsiom Yautsiukhin

Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche
Via G. Moruzzi 1, Pisa 56124, Italy
{firstname.lastname}@iit.cnr.it

Abstract. In this paper, we consider a policy engineering problem for attribute-based access control. The general goal is to help a policy writer to specify access control policies. In particular, we target the problem of defining the values of attributes when access to an object should be granted or denied. We use risk to quantify possible harm caused by misuses and abuses of granted access rights and apply the risk-benefit analysis to maximize the profit from granting an access.

Keywords: ABAC, access control, attributes, policy engineering, risk, risk-benefit analysis.

1 Introduction

Attribute-based access control (ABAC) [10, 17] is a recently proposed access-control model which generalizes the existing models such as Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-based Access Control (RBAC). Access decisions in ABAC are made on the basis of rules and policies that consist of predicates over attributes. Thus, attributes of different entities (i.e., subject, object, actions, environment, etc.) are the core of ABAC. ABAC policies no longer need ids of the entities. Instead, a policy architect should care only about relevant attributes required for policy specification.

Policies in the ABAC model may be complex and require more details than policies for other access control model. Thus, we would like to consider a policy engineering problem for ABAC to help a policy architect in and to facilitate migration of enterprises to this new model. This problem is inspired by the role engineering problem [5, 9] which aims at finding the most suitable RBAC specification.

The main policy engineering problem has several sub-problems: find the attributes required for writing a policy, assign attributes to subjects and objects, determine the exact shape of the policy, and find the values of attributes which make the policy applicable. In this paper, we would like to concentrate on the

*This work was partly supported by EU-FP7-ICT NESSoS, EU-FP7-ICT ANIKETOS, and 295354 SESAMO projects.

two latest sub-problems. We consider the case when we know which attributes are available in the system and the results of our analysis could help to define a policy and the attribute values satisfying the policy. Currently, these values are determined by experts on the basis of their experience. A more objective way of defining the policies requires the evidence of the reasons behind the policy specification.

We propose to use a risk-benefit analysis to define the values of attributes accepted by a policy. We assume that granting a permission to a user is connected with the risk that the user may misuse or abuse the obtained access rights. Thus, rules should constrain the attribute values in such a way that benefits of granting or denying access exceed the possible risk for the system. In contrast to the existing approaches which apply risk for access control [1, 4, 6, 19], we do not use risk for making access control decisions on the fly, but help to write policies in which risks are already balanced with benefits. Thus, the main contribution of the article is *a preliminary version of an approach which uses risk to specify the policy that splits the domain of values of attributes into two sub-domains*: values for which an access should be granted and values for which an access should be denied. We also show how our approach could be used in scope of bottom-up and top-down policy engineering approaches.

The rest of the paper is organized as follows. Section 2 briefly presents a statement of the policy engineering problem in the ABAC. Section 3 describes how risk-benefit analysis can be applied to solve the policy engineering problem. Section 4 discusses the preliminary results of our work. Section 5 presents the related work. The paper is concluded by Section 6.

2 Policy Engineering Problem

The ABAC is a promising approach which provides a possibility to express comprehensive access control scenarios. Though, the formal definition of the ABAC and its further configuration are challenging problems.

2.1 ABAC Core Elements

We introduce a simple ABAC model which it is inspired from the models proposed in [10, 8]. The basic elements of our ABAC model can be defined as follows:

- U is a set of *users* that issue access requests.
- O is a set of *objects* that are subject to control under security policies.
- R is a set of *actions* that can be performed on the contents of objects.
- UA is a set of names for users' attributes (e.g., "profession", "location").
- OA is a set of names for objects' attributes (e.g., "type"). All attributes in the ABAC system are denoted by the set A, $A = UA \cup OA$.
- $\mathbf{D}$ is a collection of attribute domains. Function DOM associates each attribute name with a domain of values the attribute can take, $DOM : A \rightarrow \mathbf{D}$, or simply $dom(a) = D_a$, $D_a \in \mathbf{D}$.

– Attribute assignment associates users and objects with attributes and their values, $PAV = \{(p, a, \nu) | p \in U \cup O, a \in A, \nu \in D_a \cup \{\perp\}\}$. We assign $\perp$ when the user does not posses the attribute.
– A range of the attribute $a_x \in A$ is given by $AV_x := \{(a, \nu) | \nu \in D_a \cup \{\perp\}\}$.
– A policy is a function which maps a Cartesian product of all attribute ranges and set of actions to a binary access decision, $POL : AV_1 \times ... \times AV_n \times R \to \{deny, grant\}$, and $|A| = n$. Decision making is a computation of this function.

Here we considered attributes of subjects and objects only for simplicity. Our model may be extended to use also attributes of actions, environment, etc. Taking these attributes into account will increase expressiveness of policies, but the reasoning will be the same.

Our ABAC model assumes a global security policy. However, some ABAC models, e.g., XACML OASIS standard [17], exploits rules as constructs which map attributes to a ternary access decision (grant, deny and non-applicable). Then, rules can be combined into policies, and policies can be further combined into complex policies during a computation of an access decision. All these combinations can be considered as additional assignments. In our ABAC model, we do not consider such assignments for the sake of simplicity.

2.2 Problem Statement

We consider a policy engineering problem as a task of configuring the ABAC system, i.e. defining all basic elements, *automatically*. The complexity of the policy engineering problem depends on which elements are given and which elements should be constructed or mined from the information available at the configuration time. Solving general policy engineering problem is very difficult because real ABAC implementations contain lots of functions to compose complex rules and policies, and any information in the system can be represented as an attribute.

When the policy engineering problem is defined, the algorithm which solves it should be proposed. In fact the complexity of the policy engineering problem suggests that the algorithm could provide multiple solutions of the problem. Therefore, the ABAC system architect should assess obtained solutions and pick the best one.

2.3 Our Approach

As an initial approach, we consider the policy engineering problem assuming that all elements of the ABAC system excluding POL are known. Also, we assume that there is some additional information regarding the user's behaviour after granting the access. These assumptions are applicable for systems which are configured and work properly but additional security constrains could benefit the system practicality (e.g., increase the revenue).

We use the notion of risk to tackle the problem. Moreover, risk helps choosing the best possible solution of the problem. The following running example explains our approach.

Running Example The on-line retailer (e.g., Amazon) provides the possibility to pay for items that it sells at the time of actual delivery. This kind of service is usually called as a "collect on delivery" (COD). However, the retailer may decide whether to send an item based on some attributes of the user and/or the item. These attributes might be considered as a guarantee of the eventual payment and may mitigate risks of potential fake orders and dishonest customers.

We assume the following elements in the system:

– U is a set of customers, O is a set items, d is an action "delivery" (i.e., $R = \{d\}$).
– $UA = \{P, L, Y\}$ is a set of user's attributes, where P encodes user's profession and the domain of P is $D_P = \{Student, Engineer, Manager\}$, L states the location and $D_L = \{Livorno, Lucca, Pisa\}$, Y is the user's age and $D_Y = \{18-30, 31-45, 46-99\}$.
– $OA = \{T\}$ is a set of object's attribute, where T specifies the type of the item bought by the user and $D_T = \{Book, CD, DVD\}$.

Now the problem is to determine the policy (grant or deny delivery) based on these attributes in such a way that minimizes risks for the retailer.

3 Policy Specification with Risk

A rigorous approach is required for policy engineering to make sure that policies grant access rights only to trusted users. On the other hand, the access rights may be abused even in case of correct policy engineering. We propose to exploit a risk-based method which allows minimizing risk connected with granting and denying access.

3.1 Risk Model

We consider possible risk connected with improper use of granted access rights by a user. By improper use we mean incorrect assignment, intentional abuse or (unintentional) misuse of granted access rights. Usually, risk of an event e, i.e., $Risk(e)$, is evaluated considering the probability of the event to occur $\mathbf{Pr}[e]$ together with an outcome of the event $U(e)$ (i.e., utility). Formally:

$$Risk(e) = \mathbf{Pr}[e] \cdot U(e) \tag{1}$$

While the variables in the risk equation can be evaluated either qualitatively or quantitatively. Qualitative approach can be more practical since the evaluation of qualitative risk is an easier task, moreover qualitative values can be easier to understand by security specialists. We further follow the quantitative approach similarly to our earlier works [11, 12].

The purpose of risk in our model is to establish a mapping between attributes values and access decisions. We assume that each policy depends on several attributes of subjects, objects, and environment. The decision to grant an access depends on the values of the attributes during access request. We assume that each set of attribute values causes risk of different level to a system and the system owner. A natural decision is to allow the accesses with low risk and to forbid the accesses when the risk is high.

Suppose, there is a policy $POL((a_1, \nu_{a_1}), \dots, (a_n, \nu_{a_n}))$ that leads to granting an access right. This policy depends on attributes $a_i \in A$.

We can compute risk of granting an access for a set of attribute values. The event e in this case is a set of attribute values used for an access decision:

$$e = \{(a_1, \nu_{a_1}), \dots, (a_n, \nu_{a_n}) | \nu_{a_i} \in D_{a_i}\} \tag{2}$$

We assume that there is a function $\mathbf{Pr}_{vio} : E \to [0, 1]$ which maps the set E of events e to a probability of a policy violation caused by granting access when the attributes have some certain values. There is also function $\mathbf{Pr}_{occ} : E \to [0, 1]$ which maps the set E of events e to a probability of occurrence of an event e.

Next we assume that a system owner obtains utilities granting an access. These utilities are the cost of abusing the granted access U^- (which is a negative utility) and the gain of granting access U^+ (which is a positive utility). Thus, the risk of granting the permission to a user in case of event e:

$$Risk(e) = \mathbf{Pr}_{occ}[e] \cdot \mathbf{Pr}_{vio}[e] \cdot U^- \tag{3}$$

Similar, the benefit of granting a permission in case of event e is:

$$Ben(e) = \mathbf{Pr}_{occ}[e] \cdot (1 - \mathbf{Pr}_{vio}[e]) \cdot U^+ \tag{4}$$

Where $\mathbf{Pr}_{occ}[e] \cdot \mathbf{Pr}_{vio}[e]$ and $\mathbf{Pr}_{occ}[e] \cdot (1 - \mathbf{Pr}_{vio}[e])$ are joint probabilities that an event occurs and a policy is or is not violated correspondingly.

Let $E = E^G \bigcup E^D$, where E^G is a set of events when access is granted, while events from E^D lead to the denial of access. Our goal is to split the set of events, i.e., to find the values of attributes, in such a way, that risk for the system is acceptable.

Thus, the average utility for a single access is:

$$\langle U_{E^G} \rangle = \sum_{\forall e \in E^G} \mathbf{Pr}_{occ}[e] \cdot (\mathbf{Pr}_{vio}[e] \cdot U^- + (1 - \mathbf{Pr}_{vio}[e]) \cdot U^+) \tag{5}$$

Using this general model for the average utility we discuss several strategies to mitigate the risk.

3.2 Risk Mitigation

We mitigate risk in a system by engineering ABAC policies such that only low risk accesses are granted. We suppose that the goal for the mitigation strategy is to maximize the monetary benefit. Thus, we would like to maximize the average utility. Often other constrains, except risk, are also taken into account during policy engineering. For example, the constraint may state that at least 10 users must have access. Thus, we need to solve an optimization problem [2]:

$$maximize\ \langle U_{E^G} \rangle (x) \tag{6}$$

Where x is a vector called an optimization variable of the problem, such that every element of this vector $x_i \in \{0, 1\}, i = 1, \dots, |E|$ and $x_i = 1$ if $e_i \in E^G$ and $x_i = 0$ if $e_i \in E^D$. Let $C(x)$ be any constraint function which could be bound as follows:

$$C(x) \geq 0 \tag{7}$$

Using Equations 5 and 6 we obtain the following optimization problem:

$$maximize \sum_{i=1}^{|E|} \mathbf{Pr}_{occ}[e_i] \cdot (\mathbf{Pr}_{vio}[e_i] \cdot U^- + (1 - \mathbf{Pr}_{vio}[e_i]) \cdot U^+) \cdot x_i \tag{8}$$

$$C(x) \geq 0$$

$$x_i \in \{0, 1\}, i = 1, \dots, |E|$$

The set of feasible solutions for this problem can be determined by the desire of a system owner to obtain a profit from the access in average:

$$\langle U_{E^G} \rangle > 0 \tag{9}$$

While the problem is generally NP-hard, it can be solved in polynomial time for some constraint functions $C(x)$ using linear programming approaches.

Obtain Profit from a Single Access Assume, that no additional constraints ($C(x) = 0$) are applied to Equation 9. Then, it is simply enough to balance risk and benefits for every single access in order to maximize the average utility :

$$\forall e \in E, Risk(e) + Ben(e) > 0 \tag{10}$$

It means that we select only such sets of attribute values that lead to a gain from the access rather than to a cost. Probability $\mathbf{Pr}_{occ}[e]$ does not impact whether a summand is positive or negative. Thus, we should find the threshold probability $\mathbf{Pr}^+_{vio}$ to find the set of events solving the following equation:

$$(1 - \mathbf{Pr}^+_{vio}) \cdot U^+ + \mathbf{Pr}^+_{vio} \cdot U^- = 0 \tag{11}$$

Trivially, the solution of the equation is:

$$\mathbf{Pr}^+_{vio} = \frac{U^+}{U^+ - U^-} \tag{12}$$

Note, that $U^- < 0$ because it is negative utility. Thus, $0 \leq \mathbf{Pr}^+_{vio} \leq 1$.

The set of events that grant the access is:

$$E^G = \{e : \mathbf{Pr}_{vio}[e] < \mathbf{Pr}^+_{vio}\} \tag{13}$$

The set of events that deny the access is:

$$E^D = E \setminus E^G \tag{14}$$

Running Example We continue the running example started in Section 2. We present here just some possible sets of attributes among 81 ones. There are following events that can occur during access requests:

$$\begin{aligned} e_1 &= ((P, \mathit{Student}), (L, \mathit{Pisa}), (Y, 18-30), (T, \mathit{Book})) \\ e_2 &= ((P, \mathit{Student}), (L, \mathit{Luca}), (Y, 18-30), (T, \mathit{Book})) \\ e_3 &= ((P, \mathit{Engineer}), (L, \mathit{Livorno}), (Y, 30-45), (T, \mathit{DVD})) \\ e_4 &= ((P, \mathit{Manager}), (L, \mathit{Lucca}), (Y, 46-99), (T, \mathit{CD})) \end{aligned} \tag{15}$$

Let U^- be equal for any item because a seller should only pay 7 Euros to the post for a return of the item back to the warehouse:

$$U^-(e_1) = U^-(e_2) = U^-(e_3) = U^-(e_4) = 7 \tag{16}$$

Moreover, suppose a seller obtain 0.3 Euro of gain U^+ from each successful deal:

$$U^+(e_1) = U^+(e_2) = U^+(e_3) = U^+(e_4) = 0.3 \tag{17}$$

Suppose we can obtain $\mathbf{Pr}_{vio}$ from system logs:

$$\begin{aligned} \mathbf{Pr}_{vio}[e_1] &= 0.050 \\ \mathbf{Pr}_{vio}[e_2] &= 0.030 \\ \mathbf{Pr}_{vio}[e_3] &= 0.010 \\ \mathbf{Pr}_{vio}[e_4] &= 0.005 \end{aligned} \tag{18}$$

In the example, there are no additional constrains on the policies. Thus, we may consider every single event separately as it is shown in Equation 10. Risks and benefits in this case are:

$$\begin{aligned} \mathit{Risk}(e_1) &= 0.35, \mathit{Ben}(e_1) = 0.29 \\ \mathit{Risk}(e_2) &= 0.21, \mathit{Ben}(e_2) = 0.29 \\ \mathit{Risk}(e_3) &= 0.07, \mathit{Ben}(e_3) = 0.30 \\ \mathit{Risk}(e_4) &= 0.04, \mathit{Ben}(e_4) = 0.30 \end{aligned} \tag{19}$$

From Equation 12, the threshold probability in this case is $\mathbf{Pr}^+_{vio} = 0.041$. Obviously, the user that provided the set of attributes values corresponding to the event $e_1 = ((P, \mathit{Student}), (L, \mathit{Pisa}), (Y, 18-30), (T, \mathit{Book}))$ is denied to access COD service. Users that provide the set of attributes values corresponding to events e_2, e_3, e_4 are granted with the access to COD.

4 Discussion

We would like to discuss a relation between role engineering in RBAC and risk-based policy engineering in ABAC.

There are two approaches to solve the role engineering problem [5, 9]: a top-down approach and a bottom-up approach. The top-down approach uses business-related information (e.g., hierarchy of employees, structure of the enterprise, business processes executed by the company, etc) in order to specify possible roles and subject-role (SA) and role-permission (RA) assignments. The bottom-up approach uses the information about past accesses (as logs or access-control lists) to infer the required information. The bottom-up approach that is known as *role mining* often requires automatic support.

We also may apply a similar separation of approaches for a policy engineering problem in ABAC. In Section 3 we said, that the information about the required probabilities and utilities is provided by experts. Such a top-down approach is useful when a new access control system is set up.

In contrast, if access control system is already in place, and the company would like to migrate to ABAC which is easier to manage, then the same information may be received from the history of previous accesses. In this case, we will have a bottom-up approach. We assume that there is a log containing a comprehensive information about previous accesses, similar to the assumptions made in role mining. Thus, for each abuse of a permission we can obtain an information about attribute values during the abuse. Also probability $\mathbf{Pr}_{occ}[e]$ and $\mathbf{Pr}_{vio}[e]$ could be taken from statistics. For example, if transitions from one value to another can be modeled with Markov Process, then these probabilities could be seen as steady probabilities. The utility is the business-related information and depends on possible harm to the guarded resources. If the information about the losses caused by improper usages of access rights is contained in the log, this information can be derived from the history.

On the other hand, the assumption that the information about possible losses is contained in the log is very strong. More likely, that additional information will be obtained during the operation of the system. Probabilities and utilities should be correspondingly updated as new information is obtained. Such an update may cause changes of risk and benefit and, thus, a reassignment of policies.

5 Related Work

To the best of our knowledge our work is the first one trying to formalize ABAC policy engineering problem using risk. There are several risk-based approaches tackling different issues of access control on the basis of risk.

There are approaches that enhance an access decision with risk [3, 4, 6, 11, 12, 15, 19]. Dimmock et al. [7] extend RBAC policies with risk and trust. Authors propose to use risk and trust during an access decision together with usual credential. According to the model, risky actions should be allowed only to highly trusted users. Zhang et al. [19] weight each access decision with risk and benefit and make the decision on the basis of risk-benefit analysis. Similarly, Diep et al., [6] enforced access control policies comparing risk of an action with a predefined threshold. Celikel et al. [3] introduce risk-based approach for RBAC that allows evaluating possible abuse and misuse of roles by a user in a database environment. The risk assists a database administrator to make a finer-grained decisions about granting or denying the access. Chen and Crampton [4] also consider risk as additional parameter that helps to enhance decision making in RBAC. Authors consider several factors impacting access decisions such as user's trustworthiness, degree of competence, and degree of appropriateness of user-to-permission assignment. Ni et al. [15] assume that in critical situations the access to a resource can be granted to a risky user if mitigation actions are planed in the future. In contrast to the described approaches, we use risk to engineer policies instead of enhancing accesses decisions.

Several risk-based approaches allow analyzing and managing different aspects of RBAC model [1, 13, 16]. Nissanke and Khayat [16] propose to use risk for evaluating permissions in RBAC model and then use the risk-based evaluation to manage roles hierarchy. Aziz et al. [1] propose an approach for reconfiguring RBAC policies such that risk in a system decreases. While these approaches focus on the analysis and management of deployed access control systems, our approach focuses on engineering and deploying a new attribute-based access control system.

6 Conclusion and Future Work

This paper presents the first steps towards risk-based policy engineering. We showed how a policy architect can use risk to specify the values of attributes to guarantee the least risky policy specification. As future work, we would like to consider mining probabilities and utilities from history of previous accesses and updating probabilities and utilities during system exploitation. Moreover, we are going to extend our approach to solve also the problem of identifying the attributes required for policy specification and determination of a policy shape. Also, we would like to use a more sophisticated ABAC model, e.g., Usage Control (UCON) [14, 18], which introduces mutable attributes and a continuous policy evaluation. Finally, the algorithm that solves the policy engineering problem should be capable to generate security policies exploiting existing access control languages, e.g. XACML.

References

1. B. Aziz, S. N. Foley, J. Herbert, and G. Swart. Reconfiguring role based access control policies using risk semantics. *Journal of High Speed Networks*, 15(3):261–273, 2006.
2. S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.
3. E. Celikel, M. Kantarcioglu, B. Thuraisingham, and E. Bertino. Usage control in computer security: A survey. *Risk and Decision Analysis*, 1(1):21–33, 2009.

4. L. Chen and J. Crampton. Risk-aware role-based access control. In *Proceedings of the 7th International Workshop on Security and Trust Management*, pages 140–156, 2011.
5. A. Colantonio, R. D. Pietro, A. Ocello, and N. V. Verde. A new role mining framework to elicit business roles and to mitigate enterprise risk. *Decision Support Systems*, 50(4):715731, 2011.
6. N. N. Diep, L. X. Hung, Y. Zhung, S. Lee, Y.-K. Lee, and H. Lee. Enforcing access control using risk assessment. In *Proceedings of the 4th European Conference on Universal Multiservice Networks*, pages 419–424, 2007.
7. N. Dimmock, A. Belokosztolszki, D. Eyers, J. Bacon, and K. Moody. Using trust and risk in role-based access control policies. In *Proceedings of the 9th ACM Symposium on Access Control Models and Technologies*, pages 156–162, 2004.
8. D. Ferraiolo, V. Atluri, and S. Gavrila. The policy machine: A novel architecture and framework for access control policy specification and enforcement. *Journal of Systems Architecture*, 57(4):412–424, 2011.
9. M. Frank, J. M. Buhmann, and D. Basin. On the definition of role mining. In *Proceedings of the 15th ACM Symposium on Access Control Models and Technologies*, pages 35–44. ACM.
10. X. Jin, R. Krishnan, and R. Sandhu. A unified attribute-based access control model covering dac, mac and rbac. In *Proceedings of the 26th Annual IFIP WG 11.3 conference on Data and Applications Security and Privacy*, pages 41–55, 2012.
11. L. Krautsevich, A. Lazouski, F. Martinelli, P. Mori, and A. Yautsiukhin. Integration of quantitative methods for risk evaluation within usage control policies. In *Proceedings of 22nd International Conference on Computer Communications and Networks, to appear*, 2013.
12. L. Krautsevich, A. Lazouski, F. Martinelli, and A. Yautsiukhin. Cost-effective enforcement of access and usage control policies under uncertainties. *IEEE Systems Journal, Special Issue on Security and Privacy in Complex Systems*, 7(2):223–235, 2013.
13. L. Krautsevich, F. Martinelli, C. Morisset, and A. Yautsiukhin. Risk-based auto-delegation for probabilistic availability. In *Proceedings of 4th International Workshop on Autonomous and Spontaneous Security*, pages 206–220. Springer, 2011.
14. A. Lazouski, F. Martinelli, and P. Mori. Usage control in computer security: A survey. *Computer Science Review*, 4(2):81–99, 2010.
15. Q. Ni, E. Bertino, and J. Lobo. Risk-based access control systems built on fuzzy inferences. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 250–260, 2010.
16. N. Nissanke and E. J. Khayat. Risk based security analysis of permissions in rbac. In *Proceedings of the 2nd International Workshop on Security in Information Systems*, pages 332–341, 2004.
17. OASIS. eXtensible Access Control Markup Language (XACML) Version 3.0. www.oasis-open.org/committees/xacml.
18. R. S. Sandhu and J. Park. Usage control: A vision for next generation access control. In *Proceeding of MMM-ACNS*, pages 17–31, 2003.
19. L. Zhang, A. Brodsky, and S. Jajodia. Toward information sharing: Benefit and risk access control (BARAC). In *Proceedings of the 7th IEEE International Workshop on Policies for Distributed Systems and Networks*, pages 45–53, 2006.