Improving manageability of access control policies

Jasper Bogaerts and Bert Lagaisse

iMinds-Distrinet, Department of Computer Science, KU Leuven
Celestijnenlaan 200A, 3001 Leuven, Belgium
firstname.lastname@cs.kuleuven.be

Abstract. Applications are continuously increasing in both complexity and number of users they serve. Moreover, the set of applications used by organizations is continuously expanding. This poses challenges, not in the least with regard to access control. More specifically, manageability of access control policies becomes more difficult. This leads to administrative overhead and challenges in enforcing a consistent security policy. The goal of this PhD project is to increase manageability of access control by supporting refinement of application-specific access control policies from explicitly specified organization-wide security policies. This paper provides an overview of the challenges and discusses the objectives we set in order to achieve it.

1 Introduction

Over the last years, organizations have been using an increasingly large number of applications. The applications used by organizations span from general applications which offer customer relationship management, sales and payroll support to industry-specific applications such as computer-aided design applications. Moreover, applications are becoming more complex and organizations are preparing their systems to serve both internal as well as external users. This has an impact on both scale and diversity of the user base. These trends pose challenges, not in the least for application security.

One of the techniques to enforce application security is access control. Access control regulates actions performed on objects by subjects (e.g. users). Access control is usually considered in three parts: *authentication* identifies a subject, *authorization* determines whether the subject is entitled to perform a certain action and *audit* aims at the monitoring of the performed actions.

One of the challenges of access control is manageability. Manageability of access control includes user management and management of policies. How policies are implemented is largely determined by the underlying access control model and can span from simple access control matrices to lists of complex rules.

The growing complexity and scale mentioned earlier are making manageability of access control increasingly challenging. For example, economic analysis of role-based access control – which is used extensively in practice [1] – suggests that role engineering and the mapping of permissions and users to roles remain the most significant adoption expenses for organizations [1]. Moreover, the number of applications used by organizations is continuously expanding. This makes it difficult to consistently specify an organization-wide security policy, as this security policy is left scattered amongst application-specific access control policies. Such an organization-wide policy specifies the organization's requirements with regard to access control on a high level. It may be implicitly or explicitly defined. The separated management of policies for each of the applications may lead to inconsistencies with regard to the organization-wide policy, as the high-level rules need to be translated manually to the policy of each of the applications.

In the context of this PhD project, we want to improve the manageability of access control with regard to policies and entitlements of users. To address this challenge, we propose refining organization-wide policies to application-specific policies. These organization-wide policies are explicitly specified, and can apply to several applications. This paper describes both challenges as well as objectives which will help to achieve the goal of the PhD project.

The paper is organized as follows: First, we discuss the state-of-the-art and state-of-practice in Section 2. Next, Section 3 identifies the challenges. In Section 4, we describe the objectives of this PhD project. Section 5 discusses the approach to achieve these objectives. In Section 6, we conclude the paper.

2 Background

Access control restricts the actions of subjects on objects by means of rules. Separating these rules from the application into policies can increase manageability.

This extracts the access control logic from the application logic. Ideally, the entire access control mechanism, except for the actual enforcement point, is separated from the application [2]. This offers several benefits. First, it enables application developers to focus on business logic. Second, it enables security administrators to specify access control mechanisms tailored to their needs. Third, it facilitates fully centralized management of all policies of an organization.

Besides decoupling the rules from the application they protect, manageability is also largely influenced by the access control model employed by the application. Over the last decades, several access control models have been proposed [2]. Examples include Mandatory Access Control (MAC), Discretionary Access Control (DAC) and Role-Based Access Control (RBAC). All of these models approach the manageability problem of security policies in a different manner. For example, Role-Based Access Control [3] (RBAC) provides scalable manageability by means of roles. A role serves as an indirection between subjects on the one hand, and permissions on the other. RBAC also offers support for hierarchies of roles and separation of duty concepts.

RBAC has been widely adopted for access control management [1]. However, the definition of RBAC policies can also introduce a few problems. These include a lack of expressiveness, which results in role explosion [4]. Role explosion is the rapid growth of roles related to distinct properties which are combined to obtain disjunct sets of permissions. For example, an organization which regulates access based on seniority and department would quickly experience these effects. As a result, there have been initiatives to increase management of the specified roles (amongst others [5]). Also, several attempts have been made to reduce problems related to expressiveness by means of RBAC extensions [6, 7].

Attribute-Based Access Control [4] (ABAC) generalizes these extensions. ABAC is an access control mechanism in which attributes, related to subjects, objects, actions and environment are used to limit access. These attributes can be seen as *(key, value)* pairs that can be assigned to the entities by administrators or be derived from external sources. Using attributes, ABAC can increase expressiveness by defining access control policies, which prevents role explosion [4].

3 Challenges

An organization-wide policy specifies the organization's requirements with regard to access control on a high level. It is present either explicitly (serving as a guideline for application policy specification) or implicitly (reflected only by the application policies) in the organization. As the number of applications used by an organization grows, it becomes increasingly complex to manage the organization-wide security policy, as it is left scattered amongst application-specific access control policies. This makes it harder to consistently manage organization-wide policies. For example, consistently defining an organization-wide policy that requires interim personnel to be employed for at least a month in order to be able to modify anything in any application becomes more difficult as the number of applications increases. Also, special access control concepts such as separation of duty over several applications can become more difficult to specify consistently when the number of applications grows.

As discussed previously, ABAC can enable organizations to define more expressive policies. As a result, ABAC can reduce role explosion [4], which can be a strain on access control manageability. However, it may also reduce manageability [8]. The administrative simplicity with respect to RBAC is quickly lost when policies involve more attributes. It also becomes more difficult to inspect the permissions of a certain subject [9].

Hence, there is a need for high-level abstractions over the attribute-based policies, offering better manageability over the security policy. These high-level abstractions should support the definition of a policy that spans over several applications, such as restricting access of interim employees during the first month. This involves the separation of the policy from the application that enforces it.

4 Objectives

In order to tackle the challenges described above, we propose two objectives: exploring management possibilities through the definition and refinement of organization-wide policies and the mapping of these policies to applications.

4.1 Refinement of cross-application policies

Abstractions over access control policies can be achieved by means of organization-wide security policies which are organized using the organizational structure. In order to enforce them, these policies need to be refined to application-specific policies. Enterprise-wide RBAC [10] took a similar approach for RBAC by means of parametrized roles on an organizational level which indicate the entitlements of users at application level. Other related work includes [11], which introduced role assignments based on business processes. However, these works focussed on RBAC. In our work, we will support attribute-based policies.

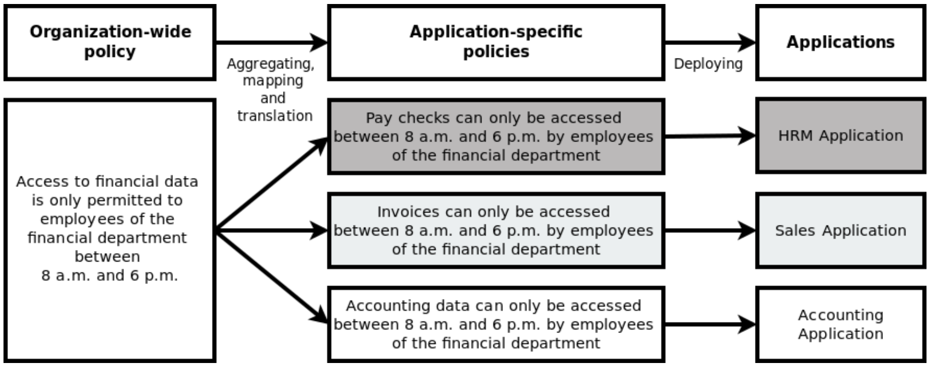
Organization-wide policy
Aggregating, mapping and translation
Application-specific policies
Deploying
Applications
Access to financial data is only permitted to employees of the financial department between 8 a.m. and 6 p.m.
Pay checks can only be accessed between 8 a.m. and 6 p.m. by employees of the financial department
Invoices can only be accessed between 8 a.m. and 6 p.m. by employees of the financial department
Accounting data can only be accessed between 8 a.m. and 6 p.m. by employees of the financial department
HRM Application
Sales Application
Accounting Application

Fig. 1: Organization-wide policy refinement requires aggregating, mapping and translating the security policy into application-specific policies. These may then be deployed at the application, or at a central policy evaluation and decision system.

Figure 1 illustrates how policies can be refined and deployed. The refinement of organization-wide policies to application-specific policies is done by first aggregating the policies for each application, mapping them to the application-specific security model and then doing the actual translation. Later, the resulting policies can be deployed at the application if required.

For example, consider the following organization-wide rule:

Access to financial data is only permitted to employees of the financial department between 8 a.m. and 6 p.m.

This rule is translated to the application-specific policies of all applications which handle financial data. Figure 1 shows the translation for the given example.

This introduces research problems with regard to how to map and refine the organization-wide policies to the application-specific policies. In this PhD project, we intend to develop techniques to refine attribute-based policies defined in the context of existing high-level concepts, such as business processes, into application-specific policies. Unlike related work [12, 13], we will focus on refinement of organization-wide policies to be enforced by access control mechanisms employed by the targeted applications, as opposed to intercepting requests or focussing on access control enforcement in general purpose programming languages. As a consequence, the organization-wide policy may be more expressive than the application-specific policies. How we can maximize expressiveness in different models with regard to attribute-based policies is another research topic that will be analysed in this PhD project. By making the right high-level abstractions to support a organization-wide security policy, we believe that policy management can be improved.

4.2 Policy-to-application mapping

As mentioned above, an important challenge in coping with policy abstraction is comprehending how to map organization-wide policies onto the security models of the targeted applications. The concepts employed in the organization-wide policies need to be mapped to application-specific policies. Target applications should supply a security model to achieve this.

In order to support automated refinement, these models should be represented in a unified way. A meta-model which enables models to reflect interrelation between their object types and actions needs to be developed. This meta-model should also reflect the subject structure of the application (e.g. the roles it provides or attributes it uses). A meta-model also enables additional techniques that improve management in access control, such as policy gap analysis.

For example, in order to determine what constitutes access to financial data in the previous example, the security model of a HRM application needs to specify that it handles pay checks and that they should be classified under financial data. In order to make organization-wide policies enforceable, the security model needs to specify which actions need to be restricted on the pay checks as well.

Previous works focussed on modelling of access control aspects in applications for testing purposes [14] or for policy specification [15]. However, they did not focus on classification of object types based on their attributes. Also, they did not focus on the mapping required to refine high-level policies based on them.

5 Approach

We intend to approach the objectives by first performing case studies on (a) a document processing platform and (b) an automated workflow platform. The analysis of the security model of both case studies provides a useful insight into the complexity of the application-specific policies that we intend to abstract to. We will leverage on this analysis to specify a generic solution which supports abstraction over all applications.

Our goals will be validated by the specification of organization-wide policies which are refined to application-specific policies. Next, we will perform a thorough evaluation on the result. A first evaluation will measure the effort that is needed in order to reuse the organization-wide security policy for different applications. This explores how much additional configuration is necessary in order to support similar applications. Secondly, the evaluation will determine how effectively it increases manageability with regard to refinement. More specifically, we will evaluate how effectively our solution supports translation of a organization-wide security policy into application-specific policies. For example, we evaluate how organization-wide policies such as the example in Figure 1 can be defined, and how much application-specific configuration (such as setting up the mapping to the security model) this requires. As such, we can compare the administrative overhead induced by application-specific policy definition with the effort of organization-wide security policy specification. This also enables us to analyse to which extent the technique supports consistent management of security policies.

As a first step in this PhD, we have looked at how XACML policies can be refined to RBAC. Next, we will look at a generic way for representing the security model of an application with regard to access control. We then investigate how organization-wide policies can be structured to support improved management.

6 Conclusion

In this paper, we motivated the requirement for an increased manageability of access control. We introduced a series of challenges to manageability in access control and discussed the objectives to this PhD project that will address them. By achieving the provided objectives, we hope to improve the current state-of-the-art in manageability techniques. This will reduce the costs related to the manageability of both security policy as well as user management.

Acknowledgements This research is partially funded by the Research Fund KU Leuven, and by the EU FP7 project NESSoS. With the financial support from the Prevention of and Fight against Crime Programme of the European Union (B-CCENTRE).

References

1. OConnor, A.C., Loomis, R.J.: Economic Analysis of Role-Based Access Control. RTI International report for NIST (2010)
2. Samarati, P., de Vimercati, S.C.: Access control: Policies, models, and mechanisms. In: Foundations of Security Analysis and Design. Springer (2001)
3. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM TISSEC **4**(3) (2001)
4. Yuan, E., Tong, J.: Attributed based access control (ABAC) for web services. In: Proceedings of IEEE International Conference on ICWS. (2005)
5. Sandhu, R., Bhamidipati, V., Coyne, E., Ganta, S., Youman, C.: The ARBAC97 model for role-based administration of roles: preliminary description and outline. In: Proceedings of the second ACM workshop on RBAC. (1997)
6. Bertino, E., Bonatti, P.A., Ferrari, E.: TRBAC: A temporal role-based access control model. ACM TISSEC **4**(3) (2001)
7. Bertino, E., Catania, B., Damiani, M.L., Perlasca, P.: GEO-RBAC: a spatially aware RBAC. In: Proceedings of the tenth ACM SACMAT. (2005)
8. Sandhu, R.: The authorization leap from rights to attributes: maturation or chaos? In: Proceedings of the 17th ACM SACMAT. (2012)
9. Kuhn, D.R., Coyne, E.J., Weil, T.R.: Adding attributes to Role-Based Access Control. IEEE Computer **43**(6) (2010)
10. Kern, A.: Advanced features for enterprise-wide role-based access control. In: Computer Security Applications Conference, 2002. Proceedings. 18th Annual. (2002)
11. Brucker, A.D., Hang, I.: Secure and compliant implementation of business process-driven systems. In: Business Process Management Workshops, Springer (2013)
12. Verhanneman, T., Piessens, F., Win, B., Joosen, W.: Uniform application-level access control enforcement of organizationwide policies. In: Computer Security Applications Conference, 21st Annual. (2005)
13. Karjoth, G.: Access Control with IBM Tivoli Access Manager. ACM TISSEC **6**(2) (2003)
14. Xu, D., Thomas, L., Kent, M., Mouelhi, T., Le Traon, Y.: A Model-based Approach to Automated Testing of Access Control Policies. In: Proceedings of the 17th ACM SACMAT. (2012)
15. Busch, M., Koch, N., Masi, M., Pugliese, R., Tiezzi, F.: Towards Model-driven Development of Access Control Policies for Web Applications. In: ACM Proceedings of MDsec. (2012)