The Challenge of Access Control Policies Quality

ELISA BERTINO and AMANI ABU JABAL, Purdue University
SERAPHIN CALO and DINESH VERMA, IBM TJ Watson Research Center
CHRISTOPHER WILLIAMS, The Defence Science and Technology Laboratory, UK

6

Access Control policies allow one to control data sharing among multiple subjects. For high assurance data security, it is critical that such policies be fit for their purpose. In this paper we introduce the notion of "policy quality" and elaborate on its many dimensions, such as consistency, completeness, and minimality. We introduce a framework supporting the analysis of policies with respect to the introduced quality dimensions and elaborate on research challenges, including policy analysis for large-scale distributed systems, assessment of policy correctness, and analysis of policies expressed in richer policy models.

CCS Concepts: • **Security and privacy → Access control**;

Additional Key Words and Phrases: Access control policies, policy analysis, provenance

ACM Reference format:
Elisa Bertino, Amani Abu Jabal, Seraphin Calo, Dinesh Verma, and Christopher Williams. 2018. The Challenge of Access Control Policies Quality. *J. Data and Information Quality* 10, 2, Article 6 (September 2018), 6 pages.
https://doi.org/10.1145/3209668

1 INTRODUCTION

Access control is a fundamental building block for secure information sharing [1]. It has been widely investigated and several access control models have been proposed, including models taking into account time, location, and situation [2, 3, 15, 16] and models specific for privacy-sensitive data [4]. Access control mechanisms are embedded in many different systems, ranging from operating systems to database management systems, and standards have been proposed—the most notable being the role-based access control (RBAC) model [5] and the extensible access control markup language (XACML) attribute-based access control model [6].

A critical issue in an access control system is represented by the access control policies. Such policies specify which subject (e.g., human user, process, application) can access which protected resources (e.g., files, database relations) for performing which actions (e.g., read, write) and they

The work reported in this article has been partially supported by NSF under grants IIS-1636891 and ACI-1547358, and by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation hereon.
Authors' addresses: E. Bertino and A. A. Jabal, CS Department, Purdue University, West Lafayette IN, 47907, USA; emails: {bertino, aabujaba}@purdue.edu; S. Calo and D. Verma, IBM TJ Watson, 1101 Kitchawan Rd, Yorktown Heights, NY 10598, USA; emails: {scalo, dverma}@us.ibm.com; C. Williams, The Defence Science and Technology Laboratory, Porton Down, Wiltshire, UK SP40JQ; email: CWILLIAMS@mail.dstl.gov.uk.

https://doi.org/10.1145/3209668

are thus the basis for access control decisions. It is obvious that a critical requirement for ensuring a correct enforcement of access control with respect to organizational security policies is that policies be of "good quality." As discussed by Bertino et al. [7], policies of low quality may lead to situations in which a party, rightfully requiring a data item, is denied access to the data item, or the data is released to unauthorized parties. In other cases, the access control system may not even have a policy concerning certain requests which, depending on the specific access control mechanism used, may lead to uncertainties concerning the outcome of the request.

In this article, we first briefly discuss relevant data quality requirements related to policies. We then present an initial infrastructure for the analysis of access control policies and elaborate on a few research challenges.

2 QUALITY REQUIREMENTS FOR POLICIES

The problem of assuring the quality of a set of access control policies can be restated as the problem of making sure that the set of policies is consistent, relevant, minimal, and complete with respect to the actions to be executed by the subjects. In what follows, we elaborate on those basic requirements.

Consistency is critical for access control systems that support denials (also called negative policies) in addition to permissions (also called positive policies). A permission states that a subject is allowed to perform a given action on a given object—for example, permission p=<*Bob, read,* +, F_1> states that *Bob* can read file F_1; whereas, a denial explicitly states that a subject is not permitted to perform a given action on a given object—for example, denial p'==<*Bob, write,* -, F_1> states that *Bob* cannot write file F_1. Denials have been introduced to support exceptions—for example, to support a policy by which a subject is allowed to read all the files in a directory except for one specific file. Thus, they have been introduced in access control mechanisms for systems where protected resources are organized according to hierarchies—notable examples are represented by access control mechanisms for object-oriented database systems [8] and the access control mechanism of the Microsoft SQL Server. Consistency thus requires that the set of policies does not include both positive and negative policies assigned to the same subject for the same action on the same object. Such conflicts are often referred to as *modality conflicts* [14].

Relevance requires that the set of policies does not contain policies that do not apply to any action executed by subjects. Irrelevant policies may undermine security. For example, an attacker may try to compromise a subject in order to exploit the permissions of this subject. Thus, making sure that subjects do not have permissions for accesses that they are not expected to execute minimizes the risk of such exploitations.

Minimality refers to making sure that the set of policies does not include redundant policies. For example, consider a file directory and a policy stating that a subject can read all the files in the directory. Then, assigning the same subject a read permission on each single file in the directory would result in a policy set with redundant policies. Redundant policies increase the administrative work required to manage policies; for example, if a subject is not any longer required to access a given object, all the policies covering such access must be properly modified and/or removed, thus increasing the security risk if these revocations are not properly executed.

Completeness requires that for any action to be executed by the subjects of the system there is a corresponding policy controlling such execution requests. If for a given access request, there is no corresponding policy, the default decision usually taken by access control systems is to deny the access. Such an approach, however, may lead to situations in which subjects that have a legitimate reason for accessing the protected resource are denied access, which may adversely affect the subject tasks/missions.

Correctness refers to making sure that the policies comply with their intended goals. Correctness of a set of policies is typically validated against a set of semantic properties that are application dependent. As discussed by Martin et al. [12], verifying policy correctness requires identifying discrepancies between the policy specifications and their implementation in a given language, such as, for example, XACML, with respect to a given set of semantic properties. Addressing correctness also requires that policies are in compliance with the system requirements [14].

In addition to those basic requirements, there are two additional complementary requirements that focus more on usability of policies. The first such requirement is policy *enforceability*; it refers to whether a policy can actually be enforced in a certain context and at which cost and with which latency. For example, a policy may require contextual information to be acquired in real time and it is crucial to provide indicators about the feasibility of acquiring such information. It is clear that, if a set of policies has poor enforceability, it will not be usable in automatically determining whether requested accesses can be permitted or not, thus basically requiring human intervention for controlling access. Assessing policy enforceability is particularly critical for novel distributed environments such as edge computing environments [20].

The second usability requirement is policy *explainability*; it refers to: (i) the ability to abstract policies at multiple levels so that human users can understand the policies that are in place; (ii) the ability to explain to human users access control decisions about access control—this is relevant especially for large policy sets when policies also include contextual conditions, such as conditions on time [2], location [3], situation [15, 16], and trust [17, 18].

3 AN INFRASTRUCTURE FOR POLICY ANALYSIS

To date, several methods for policy analysis have been proposed [9]. However, such previous methods have two major drawbacks: they focus on a single quality requirement (for example, consistency); and they require as input the specification of all possible access control requests. In particular, the latter drawback makes such previous methods not suitable for mobile distributed systems, such as mobile autonomous Internet of Things (IoT) systems and next-generation coalition systems [10], in which it is not often possible to determine in advance all actions that will have to be executed. Therefore, it is not possible to statically analyze the quality of the policies. We need an approach by which policies are analyzed at "runtime" based on information on the actual behavior of subjects in the system.

An initial infrastructure supporting policy analysis services has been recently proposed [7]. The infrastructure (see Figure 1) supports analysis with respect to four of the basic requirements described in the previous section, namely: consistency, relevance, minimality, and completeness.

The infrastructure leverages data provenance that refers to historical records about data objects. Such records typically include activities and contexts leading to the current states of the data objects of interest [7]. Data provenance thus includes all actions executed on the data objects of interest; tracking all such actions, through a provenance management system, is critical for the assessment of certain quality requirements as discussed below. In our infrastructure, we use the SimP provenance management system [11] that also records which specific policies have been applied for deciding about access requests to specific data objects. However, other provenance systems with similar functions could also be used.

The infrastructure is able to support two types of analysis: analyses that only require as input the set of policies (e.g., analysis of consistency and minimality), and analyses that also require as input information about the actions executed by the subjects in the system (e.g., analysis of completeness and relevance). The latter type of analysis is based on log provenance information about the transactions executed in the system from which access patterns are extracted and

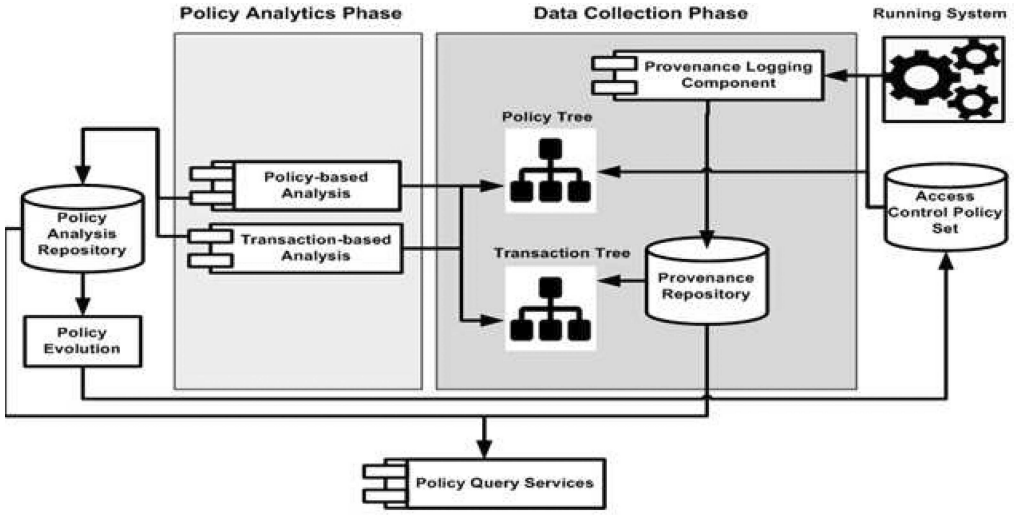

Fig. 1. A provenance-based infrastructure for policy analysis [7].

compared against the set of policies. Such extraction is performed at runtime and, thus, allows one to compare the policies against the actual behavior of subjects in the system.

The infrastructure has, however, several limitations; addressing such limitations requires addressing several challenges that we discuss in what follows.

Support of richer access control models. The infrastructure only supports RBAC policies. A major research direction is to design infrastructures supporting comprehensive analysis of complex access control models, such as attribute-based, context-based, and situation-based access control models. Such models provide two main extensions with respect to RBAC and older models: (a) support for access control decisions that take into account information about contexts and situations; and, (b) attribute-based specifications of subjects and objects in access control policies. Considering information about contexts and situations in addition to information about subjects, protected objects, and actions, allows one to specify permissions that can only be used in certain contexts or situations; an example is a permission stating that a given subject can read a sensitive dataset only when the subject is in a classified facility. Attribute-based specifications of subjects and objects allows one to specify subjects and objects as Boolean combinations of predicates against attributes characterizing properties of subjects and objects. An (informal) example of an attribute-based access control policy is one stating that all researchers that have worked over the past 5 years on projects by the crypto division are allowed to read the restricted news file. Such a policy gives permission to read the file to every researcher that verifies the predicate stating that researchers have worked over the past 5 years on projects by the crypto divisions.

Supporting analyses for such access control models requires the ability to collect information not only about transactions but also about contexts and situations in which transactions are executed, which, in turn, requires suitable secure provenance systems. In addition, when dealing with attribute-based access control it is critical that the attributes of subjects and objects be of high quality (according to the quality dimensions identified for data and information [22]). This implies that good quality policies, in turn, require that all the information needed by access control decisions be of good quality.

Another technical issue concerning the analysis of attribute-based access control policies is related to consistency. Consistency requires assuring that the predicates in the policies that allow

access and the predicates in the policies that do not allow access do not both hold true from any given access request. It is clear that whether the predicates hold true depends on the subjects and objects involved in the submitted requests. As the set of possible requests is usually not known in advance, it may not be always possible to statically determine whether a set of policies is consistent. In some cases, statically one can only determine that a set of policies may have potential inconsistencies [19]. Such analysis techniques, thus, need to be extended by the use of additional information concerning the subject population and the set of protected resources in the system of interest, if available. Otherwise, the analysis has to be completed at runtime based on data about the actual population of subjects and objects in the system.

Finally, one last research issue is related to the privacy of subject and object attributes. This issue is relevant for cloud systems and edge computing systems as attribute-based access control may require sending the cloud or some edge server such attributes and, thus, the cloud/edge server may learn private information about subjects and objects. Techniques have been proposed addressing such issues for cloud systems [21] and need to be extended for application in edge computing [20]. With respect to policy quality, the key challenge for cloud systems and edge computing is how to support policy quality analysis when policies and/or transactions and/or subject and object attributes are private.

Assessment of policy correctness. Policy correctness has been widely investigated [12, 13] and methodologies have been proposed to better align the access control policies with system requirements [14]. The main challenge in assessing correctness is when the systems of interest continuously evolve (e.g., such as in the case of adaptive systems). For such systems, the properties against which the policies are verified may have to evolve as well. Addressing such challenge requires the analysis infrastructure to assess the system state against the properties to determine whether the properties need to evolve and, if this is the case, to re-assess the policies affected by the evolved properties.

Assessment of policy enforceability. The design of methods to assess the enforceability of a set of policies is challenging as enforceability is very much context-related; for example, a policy, which can be easily enforced in a conventional enterprise setting, may not be enforceable in an edge-computing context. One possible approach to address such issues is to test policies with respect to a set of scenarios to determine their enforceability. Notice that such sets of scenarios may not cover all possible policy deployment scenarios. Also the system of interest (e.g., the system managed by the policies) may continuously evolve. Therefore, assessing enforceability of a set of policies also requires monitoring the system of interest to gather information about failures/delays due to difficulty or impossibility of enforcing certain policies. Depending on such failures/delays, one may have to evolve the policies and/or design contingency plans to deal with policies that cannot be enforced or have a very high time overhead.

Automatic policy evolution. An interesting research challenge is related to automatic policy evolution based on the results of policy analysis as, when dealing with a large set of policies deployed in highly dynamic contexts, it is likely that policies will have to continuously adapt. Supporting automatic policy evolution requires, in turn, techniques for change impact assessment [13].

Policy analysis in large-scale distributed systems. The issue is whether policy analysis (and possibly consequent policy evolution) must be carried out centrally or locally at different subsystems. Ensuring that a set of policies is of high quality at a global level may be difficult if at all possible, as different portions of the system may be characterized by different access patterns and different contexts. On the other hand, carrying out policy analysis according to a distributed strategy so that policies are analyzed at different subsystems may result in policy analysis and

evolution that are optimal with respect to local contexts but not optimal at a more global level. A possible approach is to have a flexible analysis infrastructure able to support and possibly combine both approaches depending on the specific requirements of the system of interest.

REFERENCES

[1] E. Bertino, G. Ghinita, and A. Kamra. 2011. Access control for databases: Concepts and systems. *Found. Trends Databases* 3, 1–2 (2011), 1–148.

[2] E. Bertino, P. A. Bonatti, and E. Ferrari. 2001. TRBAC: A temporal role-based access control model. *ACM Trans. Inf. Syst. Secur.* 4, 3 (2001), 191–233.

[3] M. L. Damiani, E. Bertino, B. Catania, and P. Perlasca. 2007. GEO-RBAC: A spatially aware RBAC. *ACM Trans. Inf. Syst. Secur.* 10, 1 (2007), 2.

[4] Q. Ni, E. Bertino, J. Lobo, C. Brodie, C.-M. Karat, J. Karat, and A. Trombetta. 2010. Privacy-aware role-based access control. *ACM Trans. Inf. Syst. Secur.* 13, 3 (2010), 24:1–24:31.

[5] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. 1996. Role-based access control models. *Comput.* 29, 2 (1996), 38–47.

[6] OASIS. Extensible access control markup language (XACML), *Version* 2.0 (2005). https://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

[7] E. Bertino, A. A. Jabal, S. Calo, C. Makaya, M. Touma, D. Verma, and C. Williams. 2017. Provenance-based analytics services for access control policies. In *Proceedings of 2017 IEEE World Congress on Services (SERVICES'17)*.

[8] F. Rabitti, E. Bertino, W. Kim, and D. Woelk. 1991. A model of authorization for next-generation database systems. *ACM Trans. Database Syst.* 16, 1 (1991), 88–131.

[9] A. A. Jabal, M. Davari, E. Bertino, C. Makaya, S. Calo, D. Verma, A. Russo, and C. Williams. 2018. Techniques for policy analysis and validation. *January 2018, submitted for publication.*

[10] E. Bertino, S. Calo, M. Touma, D. Verma, C. Williams, and B. Rivera. 2017. A cognitive policy framework for next-generation distributed federated systems: Concepts and research directions. In *Proceedings of 37th IEEE International Conference on Distributed Computing Systems (ICDCS'17)*.

[11] A. Abu Jabal and E. Bertino. 2016. SimP: Secure interoperable multi-granular provenance framework. In *Proceedings of the 2016 IEEE 12th International Conference on e-Science*.

[12] E. Martin, J. H. Hwang, T. Xie, and V. Hu. 2008. Assessing quality of policy properties in verification of access control policies. In *Proceedings of the 24th Annual Computer Security Applications Conference (ACSAC'08)*.

[13] K. Fisler, S. Krishnamurthi, L. A. Meyerovich, and M. C. Tschantz. 2005. Verification and change-impact analysis of access-control policies. In *Proceedings of the 27th International Conference on Software Engineering (ICSE'05)*.

[14] Q. He and A. I. Anton. 2009. Requirements-based access control analysis and policy specification (ReCAPS). *Inf. Softw. Technol.* 51, 6 (2009), 993–1009.

[15] A. S. M. Kayes, J. Han, and A. Colman. 2015. An ontological framework for situation-aware access control of software services. *Inf. Syst.* 53, Issue C (2015), 253–277.

[16] S. Yau and J. Liu. 2007. A situation-aware access control based privacy-preserving service matchmaking approach for service-oriented architecture. In *Proceedings of the 2007 IEEE International Conference on Web Services (ICWS'07)*.

[17] R. Bhatti, E. Bertino, and A. Ghafoor. 2004. A trust-based context-aware access control model for web-services. In *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*.

[18] M. Toahchoodee, R. Abdunabi, I. Ray, and I. Ray. 2009. A trust-based access control model for pervasive computing applications. In *Proceedings of the 23rd Annual IFIP WG 11.3 Working Conference*.

[19] D. Lin, P. Rao, E. Bertino, J. Lobo, and N. Li. 2010. EXAM: A comprehensive environment for the analysis of access control policies. *Int. Inf. Secur.* 9, 4 (2010), 253–273.

[20] P. Zhang, J. Liu, R. Yu, M. Sookhak, M. H. Au, and X. Luo. 2018. A survey on access control in fog computing. *IEEE Commun. Mag.* 56, 2 (2018), 144–149.

[21] M. Nabeel and E. Bertino. 2014. Privacy preserving delegated access control in public clouds. *IEEE Trans. Knowl. Data Eng.* 26, 9 (2014), 2268–2280.

[22] C. Batini and M. Scannapieco. 2016. *Data and Information Quality – Dimensions, Principles and Techniques*. Springer.

Received January 2018; revised April 2018; accepted April 2018