PBAC4M: Provenance-Based Access Control for Mobile*

Anne-Marie Lesas
Aix-Marseille Université, CNRS, ENSAM,
Université de Toulon, LSIS UMR 7296,
13397, Marseille, France
anne-marie.lesas@lsis.org

Omar Boucelma
Aix-Marseille Université, CNRS, ENSAM,
Université de Toulon, LSIS UMR 7296,
13397, Marseille, France
omar.boucelma@lsis.org

Julien Lacroix
Aix-Marseille Université, CNRS, ENSAM,
Université de Toulon, LSIS UMR 7296,
13397, Marseille, France
julien.lacroix@lsis.org

ABSTRACT

With the advent of mobiquitous technologies, security became a critical concern for digital services providers who wish to control access to their resources deployed in distributed and mobile environments. Traditional access control methods are no longer sufficient to protect themselves from hacking attempts to which resources are exposed in these open environments. We propose a complementary new approach based on *Provenance* which makes it possible to check that a request for access to a digital resource is compliant with the predefined use case and access rules.

CCS CONCEPTS

• **Security and privacy** → Systems security; Mobile platform security; Trusted computing

KEYWORDS

Access Control, Mobile Computing, Provenance, Secure Transactions

1 INTRODUCTION

Traditional access control (AC) methods do not take into account the antecedence of events that led to a resource access request:

- Mandatory AC is based on the confidentiality level of the resource (secret, confidential, private, public, etc.); this AC method is usually implemented into private network areas of companies for team-based access level,
- Discretionary AC enables an owner of the resource to manage users' authorizations; usually, this is done by an AC list making the mapping between the users and the actions they are authorized to do on a given resource (e.g.; read, write, update, delete, execute), an example of implementation is the AC find in Microsoft Windows operating systems (OS),
- In Role-Based AC (RBAC) scheme, the access rights are defined for the class of the user (e.g.; administrator, user, guest); this is typically the model implemented to manage the access to databases and web services,
- At the programming level, attribute-based AC can be used; the modifiers define the access level (i.e. public, protected, private, default) of the other software components (i.e. class, package, subclasses, etc.).

Earlier rule-based AC methods allow the definition of authorization rules that are checked upon an access request to a resource, and some others are based on the locality. But none of these methods enable a high-level of AC design according to predefined functional use cases. Furthermore, traditional AC methods do not address the traceability of transactions and the activities history. The provenance-based AC for mobile (PBAC4M) we propose addresses at least two distinct security-related issues [1] faced in the distributed environments, especially on mobile platforms and electronic transactions involving several stakeholders (i.e. services providers and third-parties), such as contactless payment, that require trust and strengthened security:

1. Provenance-based action validation, and
2. Provenance-based policy retrieval.

In this paper we introduce our proposal for PBAC4M.

2 RELATED WORK AND CONTRIBUTIONS

PBAC4M is the continuation of our previous work on provenance-based AC (PBAC) and the extension of well-known RBAC mechanisms with a set of rules (policy) encoded *à la Datalog* we process with a basic inference engine that acts as a mediator between different stakeholders [2]. PBAC4M implementation relies on the W3C provenance (PROV) standard [6] and the PROV Data Model (PROV-DM) [2][5] to provide, by design, a provenance check and traceability solution for digital objects (e.g.; a token) in distributed systems. Our contribution is:

- *An issue to solve*: AC strengthening in distributed and mobile environments,
- *A solution proposal*: an additional AC method with our PBAC4M model which enables the verification of compliance with predefined use case and inference rules and provides the trace of activities,
- *A proof-of-concept*: with the implementation of PBAC4M tracking system and an inference engine, which we refer as the mediator agent, for the check and the approval or denial of activities requesting the access to digital resources.

MobiQuitous 2017, November 7–10, 2017, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5368-7/17/11.
https://doi.org/10.1145/3144457.3144513

3 PBAC4M IMPLEMENTATION

PBAC4M is intended to be part of the mobile OS; each time an activity is invoked from the interface, PBAC4M engine checks if the agent associated to the requesting activity is authorized to invoke this activity for the use of a given entity (i.e. resource). Then, the inference engine checks the compliance with the predefined use case and rules of PROV-DM graph generated by the activities history. If the compliance is successfully checked, then, the inferences engine authorizes the processing of the requested activity, else, it is denied (see Fig. 1 and Fig. 2).

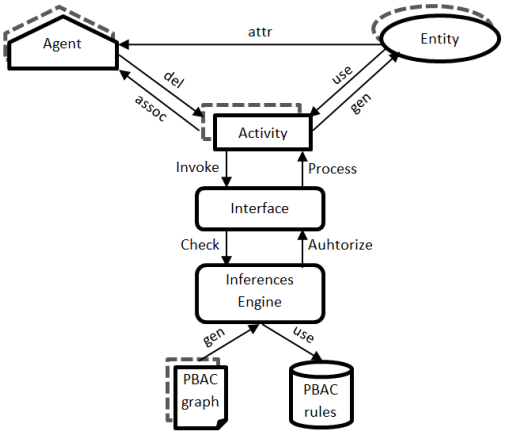

Figure 1: PBAC4M inference engine.

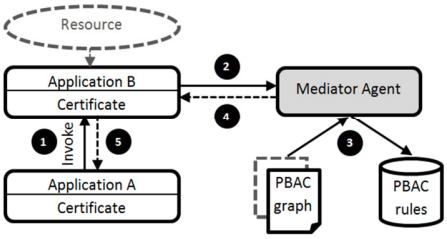

Figure 2: Generic implementation of the PBAC4M.

However, in open environments there is no guaranty on the digital resources reliability (e.g., was the resource accessed or modified by a non-trusted activity/agent before?). This is why PBAC4M is ideally suited for trusted execution environments; for instance, on recent devices, the OS is split into two areas: the "normal" Rich Execution Environment (REE) where any application can access to the shared features and resources and the tamper-resistant Trusted Execution Environment (TEE) [4] which access is restricted to the authorized and authenticated activities according the TEE AC rules. Fig. 3 illustrates an access request ❶ to a trusted activity made from the TEE API by an application running in the REE that is checked ❷ by the AC Agent: in addition to the already AC management done to access the TEE resources, PBAC4M mediator checks the PROV-DM graph of the activity history and its compliance with the PBAC rules ❸; then, the authorization result is returned to the AC agent which can authorize ❹ or deny the activity processing and the use of the resource ❺.

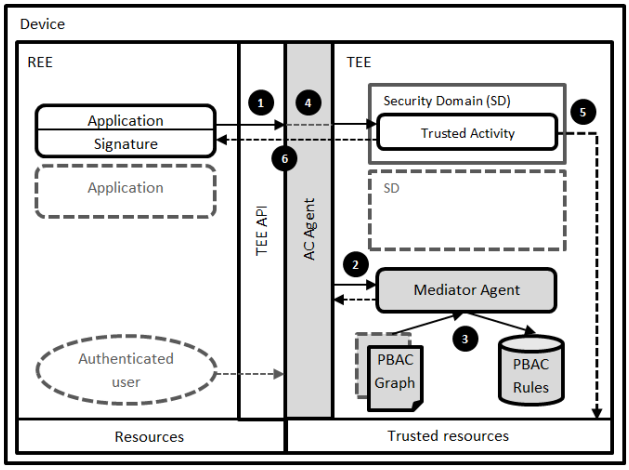

Figure 3: PBAC4M mediator agent of the TEE.

4 CONCLUSION AND FUTURE WORK

PBAC4M is a complementary AC for the digital resources hosted into the mobile devices; our solution enables to check the compliance with predefined use cases based on PROV-DM and a set of inference rules encoded *à la Datalog*. Transactions graphs generated by PBAC4M engine also provide an additional traceability. PBAC4M implementation as a TEE agent will be prototyped in a future work in partnership with industrials.

REFERENCES

[1] Jaehong Park, Dang Nguyen, and Ravi Sandhu. A provenance-based access control model. In Proceedings of the 2012 Tenth Annual International Conference on Privacy, Security and Trust, PST '12, pages 137–144, Washington, DC, USA, Jul. 2012. IEEE Computer Society.

[2] Julien Lacroix and Omar Boucelma. Trusting the Cloud: A PROV + RBAC approach. In Proceedings of the 2014 IEEE International Conference on Cloud Computing, CLOUD '14, pages 652–658, Washington, DC, USA, Jun. 2014. IEEE Computer Society.

[3] Luc Moreau and Paolo Missier. PROV-DM: The PROV data model. W3C recommendation, World Wide Web Consortium, Apr. 2013.

[4] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid. Bouabdalla, Trusted Execution Environment:What It Is, and What It Is Not, 14th IEEE International Conference on Trust, Security, Helsinki, Finland, 2015.

[5] The open provenance model: An overview. In Lecture Notes in Computer Science, Second International Provenance and Annotation Workshop (IPAW). Springer, June 2008.

[6] W3C. PROV-Overview: An Overview of the PROV Family of Documents, April 2013.