A Dynamic Context-Aware Provision-Based Access Control Model

K. Rafati*, N. Esfahani*, M. Amini*, A. R. Masoumzadeh*, R. Jalili*

Keywords: Context-aware, Provision-based, Dynamic Authorization, Access Control Model

Abstract: *In distributed system, especially in pervasive environments, security requirements have became inexpressible. Expressiveness of an access control system is tightly dependent to access control model. In traditional access control models, security requirements were expressed using core model terms which could be confining. This paper tries to introduce an access control mode with more expressiveness. For this purpose, we used propositional logic to describe security requirements. A formally specified access control model has been obtained. This model can address the problem of security requirement definition in pervasive environments suitably.*

1 Introduction

As time goes by, popularity of using distributed systems (e.g. internet, pervasive computing, etc.) increases. One of the fundamental aspects of these systems which determine degree of dependability is security. All aspects of the traditional security have meaning in these distributed systems; People are going to use them and embrace them as one aspect of their life. This will not happen until they can trust these systems. Usually, the Heart of traditional security mechanisms is an access control model which determines the remaining characteristics of the security mechanisms. Introducing security mechanism for distributed systems, first step is to define an access control model appropriate for them. Several access control model have been introduced so far (in fact this field is one of the oldest fields in computer science). We are going to introduce a few of them which had effects on our model. PBAC [3] (Provision Based Access Control) is trying to insert concept of the provision in its model. Provision sometimes is referred as obligation in literature. Provision is one of the fundamental requirements of an access control model for distributed systems. These systems require more dynamic access control model, provision provide them such flexibility. $UCON_{ABC}$ [6,7,8] defines a global model which is going to cover all known aspects of security. It introduces new concepts in security e.g. post authorization, context awareness and etc.

* Computer Engineering Department, Sharif University of Technology, Tehran, Iran
{rafati@ce. , esfahani@ce. , m_amini@ce. , masoumzadeh@ce. , jalili@}sharif.edu

$UCON_{ABC}$ is a very good model but unfortunately it has not been elaborated yet. It's evolving know and some things have been done [2] but this model is more conceptual than practical. On the other hand PBAC is more practical model which covers some aspects of security domain. In this paper we are going to use some aspects of $UCON_{ABC}$ model to extend the scope of PBAC model. Context awareness as one of the most demanding features of distributed systems is the most important features of this new model. It's true that the base of this model is PBAC but the limitations of this base model lead to changing it very much; the new model has used the core logic of PBAC model to define new features that are not present inside the core model.

The paper is structured as follow: First (in section 2) we present an informal description of the model, next in section 3 we'll meet the required architecture for implementing the model and finally at the forth section we'll describe each of model components formally.

2 Model Overview

In this paper we represent a formal definition for dynamic authorization based on conventional PBAC model. In order to enrich the model for today needs, we add some new features that we will describe them in following sections.

First we use subjects, objects and context attributes as factors for making access decisions. In other words, decision is made based on specification of environment, context of usage and attributes of subjects and objects.

A state in this security system consists of subjects and objects attributes along with context and environment attributes.

Each policy rule has a set of Pre-Conditions and Ongoing-Conditions. The Pre-Conditions must be satisfied at the start of the access, while the Ongoing-Conditions must be satisfied during access. Conditions are expressed as Boolean formula in propositional logic (which contains no quantifier) and atomic predicates are binary predicates that compare two attributes (example: John's classification is greater than book's classification) or compare an attribute with an attribute value (example: Alice's credit is greater than 10$).

There are also three kinds of obligations. Pre-Obligations must be performed before access taking place. Ongoing-Obligations must be performed during access and finally Post-Obligations must be performed after the completion of access. You can think of an obligation as a piece of code that must be executed.

There must also be a mechanism for considering ongoing accesses in order to check their authorization conditions and obligations. So for any ongoing access there exists a Session in the system which stores related data such as subject, object, action, Ongoing-Conditions and Ongoing-Obligations. So sessions are the key concepts of dynamic authorization. There exists a session repository in the system which keeps track of existing sessions and it is responsible for retrieval and updating of existing sessions as well as creating new sessions.

3 Architecture

We need to decouple responsibilities and introduce an architecture based on them to implement what we said in the previous sections. In the following subsections we first introduce common concepts that are used in the architecture then we would clarify two scenarios in which the decision can be made.

3.1 Overview

In the Figure 1 the components of our architecture can be seen. The Policy Decision Point (PDP) is the central part and decide if we can grant an access or not and on what

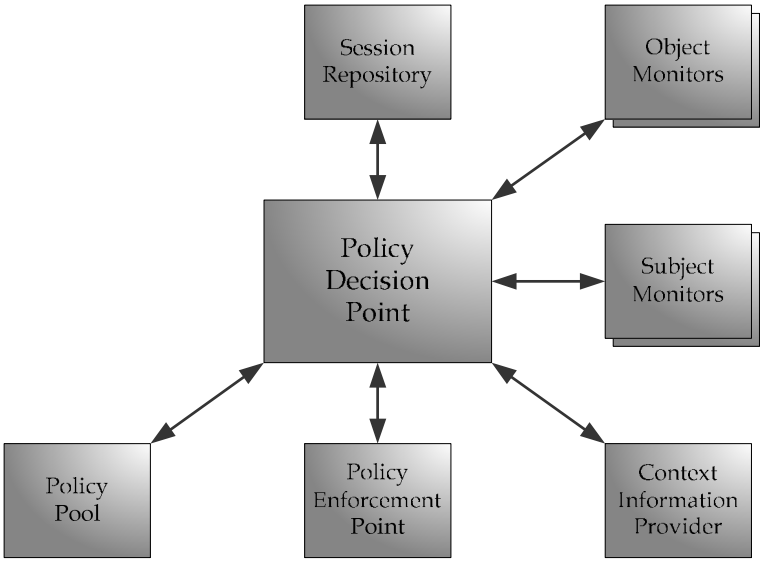

Figure 1: main components of the architecture

conditions. It also monitors live sessions for checking ongoing conditions. For deciding about accesses, PDP must have some information; this data is fed by some of the other components. Policy Pool (PP) has all the predefined policies that we must follow, Session Repository (SR) maintains the situation of the sessions, Object Monitors (OM) and Subject Monitors (SM) manage the attributes of the objects and subjects and finally Context Information Provider (CIP) is responsible for the environment and its state. We have one component left, Policy Enforcement Point (PEP) which does the PDP commands and control the enforcement of PDP decisions. PEP can also be used as a data source for PDP; it also provides PDP with the information about ongoing actions.

3.2 Request-Driven Mode

This mode of working can be seen in the Figure 2. First a decision request (DR) comes

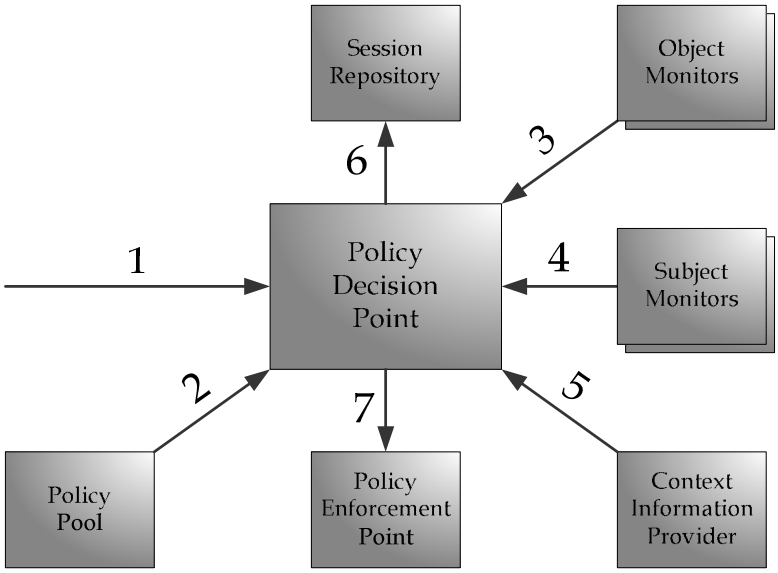

Figure 2: request-driven mode of model

(1) from an external entity to PDP. PDP retrieves the matching policies (2) to that DR from PP then it gets the needed data from OMs, SMs and CIP (3, 4, and 5). If the Policy and data match each other and access can be granted, PDP grants the access and makes a session for the access in the SR (6). Finally PSP tell PEP to enforce (7) the access and to control it. It should be noted that we have an OM and SM for each object and subject.

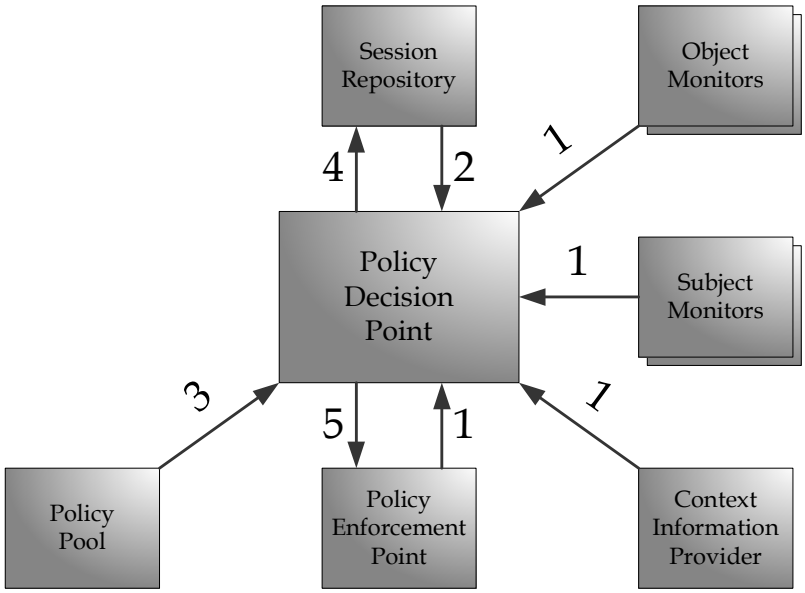

Figure 3: change-driven mode of model

3.3 Change-Driven Mode

Figure 3 represents this mode of acting. This mode along with sessions makes the model dynamic. Everything is started by a change (1) that can be seen as a trigger. This trigger comes from one of information providers of the model i.e. OMs, SMs, CIP and PEP. In fact these components tell PDP that one of the bases of it's decision has changed; OMs and SMs notify PDP about the changes in the attributes of the object which they're in charge of, CIP provides changes in the environment and PEP tell us if any ongoing provision has failed (which needs taking care and discarding the related session). After starting the PDP retrieves (2) the sessions which are related to the change. Using PP (3) it decide about retrieved sessions and delete some of them and/or create new ones and adds them to the SR (4). Finally based on the new arrangement of the sessions PDP commands (5) PEP.

4 Formal Definitions

In order to define conditions, we use propositional logic which you can see the BNF grammar of that here:

$$formula \longrightarrow < Predicate > | \left(\neg formula\right) | \left(formula \wedge formula\right)$$
$$| \left(formula \vee formula\right) | \left(formula \rightarrow formula\right)$$

Entity Attribute Set is set of all attributes of all entities in the system and is define as follow:

$$EntityAttributeSet \in EntitySet * AttributeSet$$

Predicates are used to compare entity attributes with each other and also with some attribute values:

$$Predicate \in EntityAttributeSet * Relator *$$
$$\left(EntityAttributeExpression \cup AttributeValueSet\right)$$

Conditions are sets of Boolean formulas that state conditions that must be satisfied before or during access:

$$PreConditions \in \mathrm{P}\left(formula\right)$$
$$OngoingConditions \in \mathrm{P}\left(formula\right)$$

Obligations are set of actions that must be performed before, during or at the end of access:

$$PreObligations \in \mathrm{P}\left(PvnSet\right)$$
$$OngoingObligations \in \mathrm{P}\left(PvnSet\right)$$
$$PostObligations \in \mathrm{P}\left(PvnSet\right)$$

Sessions that represent ongoing actions contain information about subject, object, action (example: read), permission (example: +), Ongoing-Conditions, Ongoing- and Post-Obligations as follow:

$$Session \in SubjectSet \times ObjectSet \times ActSet \times PrmSet \times OngoingConditions \times OngoingObligations \times PostObligations$$

Access Control Policy Rules are defined:

$$ACPR \in SH_1 \times \ldots \times SH_n \times OH_1 \times \ldots \times OH_m \times ActSet \times PreConditions \times OngoingConditions \times PrmSet \times PreObligations \times OngoingObligations \times PostObligations$$

5 Conclusion

Every computer system can be specified with attributes of its constituents. This fact plus power of propositional logic shows expressiveness of our model. In this model, we try to avoid restricting concepts such as Group, Rule, Task and etc. in order to reach a universal model. All of these concepts can be expressed as attributes. Complicated security rules and requirements of dynamic distributed environments can be easily specified using propositional logic. Enforcement and auditing policies are exerted using obligations.

Formal definition of concepts makes our model meticulous and precise. Also it has led to a straightforward implementation.

In this paper we consider obligations as a procedure to be called but more research could be conducted on obligations and their formal specification especially for post-obligations. Also an executing prototype should be developed as proof of concepts.

6 References

[1] Zhang, X., Nakae, M., Covington, M. J. and Sandhu, R., 2006. *A usage-based authorization framework for collaborative computing systems*. In Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies (Lake Tahoe, California, USA, June 07 - 09, 2006). SACMAT '06. ACM Press, New York, NY, 180-189.

[2] Zhang, X., Park, J., Parisi-Presicce, F. and Sandhu, R., *A logical specification for usage control*, In Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies (Yorktown Heights, New York, USA, June 02 - 04, 2004), SACMAT '04. ACM Press, New York, NY, 1-10.

[3] Bettini, C., Jajodia, S., Sean Wang, X. and Wijesekera, D., *Provisions and Obligations in Policy Management and Security Applications*. Proceedings 28th Conference on Very Large Data Bases, Hong Kong, China, 502–513 (2002)

[4] Lepro, R., *Cardea: Dynamic Access Control in Distributed Systems*. NASA Technical Report November 2003. (URL= http://www.nas.nasa.gov/News/Techreports/2003/2003.html) (Visit = July 2006)

[5] Jagadeesan, R. and Marrero, W., *Timed constraint programming: a declarative approach to usage control*, In Proceedings of the 7th ACM SIGPLAN international Conference on Principles and Practice of Declarative Programming (Lisbon, Portugal, July 11 - 13, 2005), PPDP '05. ACM Press, New York, NY, 164-175.

[6] Park, J. and Sandhu, R. *Towards usage control models: beyond traditional access control*, In Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies (Monterey, California, USA, June 03 - 04, 2002), SACMAT '02, ACM Press, New York, NY, 57-64.

[7] Park, J. and Sandhu, R., *the UCONABC usage control model*, ACM Trans INF Syst Secur 7, 1 (Feb. 2004), 128-174.

[8] Sandhu, R. and Park, J., *Usage Control: A Vision for Next Generation Access Control*, the Second International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security, 2003.

[9] Park, J., Zhang, X. and Sandhu, R., *Attribute Mutability in Usage Control*, 18th Annual IFIP WG 11.3 Working Conference on Data and Applications Security, 2004.