Computer Languages, Systems & Structures 49 (2017) 1–35

Contents lists available at ScienceDirect

Computer Languages, Systems & Structures

journal homepage: www.elsevier.com/locate/cl

Formal specification and integration of distributed security policies

Mohamed Mejri [a], Hamdi Yahyaoui [b,*]

[a] Computer Science Department, Laval University, Canada
[b] Computer Science Department, Kuwait University, Kuwait

ARTICLE INFO

Article history:
Received 17 May 2016
Received in revised form
22 November 2016
Accepted 22 December 2016
Available online 3 January 2017

Keywords:
Security policies
Formal languages
Semantics
Integration
XACML

ABSTRACT

We propose in this paper the Security Policy Language (SePL), which is a formal language for capturing and integrating distributed security policies. The syntax of SePL includes several operators for the integration of policies and it is endowed with a denotational semantics that is a generic semantics, i.e., which is independent of any evaluation environment. We prove the completeness of SePL with respect to set theory. Furthermore, we provide a formalization of a large subset of the eXtensible Access Control Markup Language (XACML), which is the well-known standard informal specification language of Web security policies. We also provide a semantics for XACML policy combining algorithms.

1. Introduction

Nowadays, there is a drastic growing of security threats, which benefit from security breaches in systems to jeopardize their security and achieve malicious goals such as thief and illegal access to information, identity masquerading, etc. The consequences of security attacks can be fatal to institutions and companies, which made security a major concern for people in industry and academia. In this context, building secure systems is becoming a paramount challenge mainly in a distributed environment where each system has its own security policies, which may conflict with policies of other systems. In such environment, security policies specification is based on standard languages, which are often informal and complex such as the eXtensible Access Control Markup Language (XACML) [30]; the well-known standard informal specification language of Web security policies. Such complexity makes the learning curve of the proposed languages very high and increases the likelihood of having design errors. Accordingly, there is a desideratum for providing simple and formal models that capture such policies and allow to reason about them.

There are two main classes of approaches for formalizing security policies: Model and language based methods. Model based approaches leverage formalisms such as transition systems to capture policies. Model checking of system compliance to security properties is one of the main targets behind the design of such models. The main issue with such methods is their limited scalability when applied to huge policies.

Regarding language based approaches, several languages were proposed to specify security policies. XML-based specification languages use XML tags to describe security policies and rules between subjects and resources. Famous XML-based specification languages include Security Assertion Markup Language (SAML) [22], XML Access Control Policy Specification Language (XACL) [10], and Extensible Access Markup Control Language (XACML) [30]. The issue with such languages is that they are machine readable and so difficult to be understood. Furthermore, they lack the formal aspect that allows reasoning about them. Declarative languages provide a high level of simplicity and readability for the specification of security policies. We mention Ponder [11] as a famous declarative, object-oriented language for specifying policies for the security and management of distributed systems. The main issue

* Correspondence to: Computer Science Department, Kuwait University, P.O. Box 5969, Safat 13060, State of Kuwait, Kuwait.
E-mail addresses: mejri@ift.ulaval.ca (M. Mejri), hamdi@cs.ku.edu.kw (H. Yahyaoui).

http://dx.doi.org/10.1016/j.cl.2016.12.004
1477-8424/

with such languages is the lack of abstractness that does not allow to reason about correctness and completeness issues. Event based languages, such as Policy Description Language (PDL) [9] and DEFCon Policy Language (DPL) [24], leverage events and actions to model security policies and rules between subjects and resources. Some of these languages put more focus on actions rather than data while others express constraints on event flows. The main issue with such languages is their complexity and sometimes their modeling of low level details. Algebraic languages allow to formally define security policies. An important feature of algebraic security policy languages is their simplicity, powerful expressiveness and compactness.

We advocate in this work the need for a simple (users do not need to know a heavy mathematical background to understand it), formal (precise and rigorous) and concise (compact with a short grammar) algebraic language to guarantee the absence of inconsistencies in policies, reason about their integration, and prove their correctness. To achieve this goal, we define in this paper a new language called Security Policy Language (SePL) for the specification of distributed security policies. We also show how the language can be leveraged to define the integration of policies. In addition, we present a formalization of a large subset of the latest version of XACML based on SePL, which provides a simple understanding of that language. The contributions of this paper are the following:

- The proposal of the Security Policy Language (SePL), a multivalued language for the specification and the integration of security policies.
- A BNF grammar for a large subset of XACML-3.0.
- A translation function from the XACMLs BNF grammar to SePL.
- The formalization of most of the XACML policy combining algorithms (including permit-overrides, deny-overrides, Only-one-applicable, Deny-unless-permit, Permit-unless-deny) using SePL.

The paper is organized as follows. In Section 2, we provide the background related to security policies. Section 3 is dedicated to the presentation of the syntax and semantics of SePL. We provide a formalization of XACML based on SePL in Section 4. Section 5 is devoted to the proof of completeness of SePL. In Section 6, we provide a comparison of our work with the related work. Finally, we provide some concluding remarks in Section 7.

2. Security policies

A security policy is a set of rules that define constraints on users while interacting with a system. Security policies are meant to reinforce three main properties: confidentiality, integrity and availability. Confidentiality refers to keeping the content of a communication secret. Integrity refers to keeping the content unchanged. Availability refers to the guarantee of reliable access to the information by authorized people. Different formal models for confidentiality and integrity were devised. Famous models are Bell-La Padula model [4] (confidentiality policy model) and Biba model [3] (integrity policy model).

An access control policy is related to system access rules. The rules specify who (subject) can access what (object) and under which conditions. Access control models are categorized as either discretionary or non-discretionary. There are three well known access control models: Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role Based Access Control (RBAC). DAC is a policy specified by the owner of an object. MAC refers to allowing access to a resource if there are rules which allow a user to access that resource. RBAC is an access policy that gives access rights to users based on their roles in the system.

Distributed security policies come with the concept of distribution of the policies and the decisions on the elements of a distributed system. Inconsistencies may arise due to the lack of a central entity that controls these policies. Henceforth, there is a need for checking the consistency and conformance of such type of security policies. SePL is a research initiative towards achieving this goal.

3. SePL syntax and semantics

SePL (Security Policy Language) is not intended to substitute XACML but a kind of an alternative syntax for it with a solid semantic ground. XACML is based on XML which has the benefit of interoperability, but it has many drawbacks:

- XACML is very verbose, making the specification of a simple policy long by including XML structures and by prefixing identifiers by long XACML namespaces.
- The semantics of XACML security policies becomes error prone when they involve many rules with different combining algorithms.
- Even if XACML is intended to be easily used by security managers, it is neither suitable for programmers nor for theoreticians. Programmer community has developed their own language, named ALFA [23], that is close to Java and C# to write XACML policies and developed a tool to translate formulae from ALFA to XACML. Theoreticians better prefer a concise language like SePL, where the semantics of any syntactic expression is a term of a domain that is computed by a function.

SePL provides a concise BNF syntax that is equipped with a formal semantics based on set theory. It can be leveraged to do formal verification and analysis of attribute based languages like XACML. The use of formal languages, like SePL, is helpful