Computers and Electrical Engineering 44 (2015) 91–103

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Semantics-based approach for detecting flaws, conflicts and redundancies in XACML policies ☆

Hussein Jebbaoui [a], Azzam Mourad [a,*], Hadi Otrok [b], Ramzi Haraty [a]

[a] Department of Computer Science and Mathematics, Lebanese American University, Lebanon
[b] Department of Computer Engineering, Khalifa University of Science, Technology & Research, United Arab Emirates

ARTICLE INFO

Article history:
Received 15 May 2014
Received in revised form 4 October 2014
Accepted 16 December 2014
Available online 16 February 2015

Keywords:
Web services security
Access control
Policy analysis
Set theory
Semantics
XACML

ABSTRACT

XACML (eXtensible Access Control Markup Language) policies, which are widely adopted for defining and controlling dynamic access among Web/cloud services, are becoming more complex in order to handle the significant growth in communication and cooperation between individuals and composed services. However, the large size and complexity of these policies raise many concerns related to their correctness in terms of flaws, conflicts and redundancies presence. This paper addresses this problem through introducing a novel set and semantics based scheme that provides accurate and efficient analysis of XACML policies. First, our approach resolves the complexity of policies by elaborating an intermediate set-based representation to which the elements of XACML are automatically converted. Second, it allows to detect flaws, conflicts and redundancies between rules by offering new mechanisms to analyze the meaning of policy rules through semantics verification by inference rule structure and deductive logic. All the approach components and algorithms realizing the proposed analysis semantics have been implemented in one development framework. Experiments carried out on synthetic and real-life XACML policies explore the relevance of our analysis algorithms with acceptable overhead. Please visit http://www.azzammourad.org/#projects to download the framework.

1. Introduction

The heavy reliance on Web services as one of the primary methods for data exchange between partners and distributed systems still faces the risk of exploitation as a result of their infinite accessibility over the Internet [1,2]. In addition, services with critical data such as banking and other financial businesses are emerging, which increase security challenges [3]. In this regard, policy-based computing [4–6] is taking an increasing role in governing the systematic interaction among distributed services. Particularly, access control is the most challenging aspect of Web service security to determine which partner can access which service [7]. Currently, an increasing trend is to declare policies in a standardized specification language such as XACML, the OASIS standard eXtensible Access Control Markup Language [8]. Many vendors are adopting XACML for controlling access to their services.

Before stating the addressed problems and contributions of our work, we depict in the sequel a brief introduction about XACML [8], which has a policy structure divided into three layers. The top layer consists of a policy set, the middle layer consists of policies and the lower layer consists of rules. Each layer contains a target element which is used to define the

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. Srinivasan Rajavelu.
* Corresponding author.

http://dx.doi.org/10.1016/j.compeleceng.2014.12.012
0045-7906/© 2014 Elsevier Ltd. All rights reserved.

subjects, resources and actions. The policy set contains a set of policies, a set of obligations and a policy combining algorithm used to break the tie between its policies. Each policy has a set of rules, a set of obligations and a rule combining algorithm used to break the tie between its rules. A rule consists of a set of conditions and a rule effect. The obligations at the policy set and policy level are carried out when the final decision is reached to either *permit* or *deny*. The illustrative policy set example in Fig. 1, which will be used and explained in the case study (Section 5), depicts the policy structure.

Nowadays, mid and large size online systems may embed several distributed services heavily interacting and composed together to provide features satisfying the clients' needs. This may require policies with hundreds and even thousands of rules to control access and enforce business behaviors. As a result, policies used as means of protection can be a source of weaknesses due to the presence of flaws and conflicts between their rules. For instance, considering the example in Fig. 1, rules $R3$ and $R4$ lead to an access flaw because both rules have no targets, both rules have the same effect *Permit*, $R3$ precedes in order $R4$ and $R4$ is more restricted than $R3$. With the current XACML decision mechanism, the generic rule $R3$ will always take precedence and be evaluated before the restricted rule $R4$. Therefore the response will always be given by $R3$ that grants access to any subject, while $R4$ that limits the access to subject *Joe* will be disregarded. In this context, the true objective of access control is to give higher priority to more restricted rules. Current XACML tools give major role to security administrators to resolve some tie/conflict decisions through policies/rules modifications and/or combining algorithms (e.g. $Permit - overrides$ and $First - Applicable$). Although manual corrections may seem practical for small size policies, it is doubtful if not impossible for large ones within the complex structure of XACML. The problem grows more when integrating and composing different policies [9,2,7,10,6,5], where contradictions between combining algorithms are apparent. In this regard, some approaches have been proposed addressing XACML policy composition and analysis [11–18]. However, these propositions did not address the presence of access flaws, conflicts and redundancies between policies, and did not consider the logical meaning of rules that reflect the objectives of a policy.

In this paper, we tackle the aforementioned problems by elaborating a set-based scheme that provides formal specification of policies and semantics-based detection built on top of it to efficiently perform analysis tasks. The main contributions of this paper are two folds: (1) Addressing the complex constructs of XACML through an abstract set-based syntax (SBA-XACML), while maintaining a similar policy structure that covers all its elements and sub elements and (2) offering novel detection mechanisms that analyze the meaning of policy rules through semantics verification by inference rule structure and deductive logic. All the approach components and algorithms have been implemented in one development framework that accepts XACML policies as inputs, converts them automatically to SBA-XACML constructs, and produces a list of access flaws, conflicts and redundancies between rules. The provided experiments conducted on real-life and synthetic XACML policies explore the relevance and efficiency of our analysis approach with acceptable overhead.

The rest of the paper is organized as follows. Section 2 covers for the approach overview and architecture. Section 3 presents the semantics rules for policy and rule analysis. Section 4 illustrates the analysis algorithms. Section 5 depicts the case study and semantics-based detection. Section 6 focuses on the experiments and performance analysis. Section 7 summarizes the related work. Finally, Section 8 presents the conclusion.

2. Approach Overview

The overall architecture of our approach is illustrated in Fig. 2 with all its components, i.e. SBA-XACML Language, Compiler and Analysis Module. Using the framework, the user can analyse the policies for access flaws, conflicts and redundancies and get the corresponding analysis report using the module embedding the analysis algorithms.

2.1. SBA-XACML Language and Compiler

SBA-XACML is a set-based language composed of all the elements and constructs needed for the specification of XACML based policy. Please refer to [19] for the complete definition and syntax of SBA-XACML elements and attributes. Its compiler includes XACML parser and converter to SBA-XACML. It takes XACML policy set as inputs, parses their XACML elements and generates SBA-XACML constructs according to the language syntax and structure. In the sequel, we present a brief summary about its constructs that are needed in this paper. SBA-XACML based policy, referred to as a policy set PS, is ordered into 3 levels: $PolicySet$, $Policy$, and $Rule$. Every element can contain a $Target$. $PolicySet$ element contains other $PolicySet(s)$ and/or $Policie(s)$. $Policy$ contains $Rule(s)$.

A target TR is an objective and is mapped to SBA-XACML within the context of rule, policy and policy set according to the following syntax:

$$TR = \{S, R, A\}$$

where S is a set of subjects, R is a set of resources and A is a set of actions.

PS may contain other policy sets, policies or both. It can also be referenced by other policy sets. It is mapped to SBA-XACML according to the following syntax:

$$PS ::= \langle ID, SP, PR, PCA, IPS, OBLs, TR\rangle$$