Chapter 4
A JSON-Based Fast and Expressive Access Control Policy Framework

Hao Jiang
New H3C Technologies Co. Ltd., China

Ahmed Bouabdallah
IMT Atlantique, France

ABSTRACT

Along with the rapid development of ICT technologies, new areas like Industry 4.0, IoT, and 5G have emerged and brought out the need for protecting shared resources and services under time-critical and energy-constrained scenarios with real-time policy-based access control. To achieve this, the policy language needs to be very expressive but lightweight and efficient. These challenges are investigated and a set of key requirements for such a policy language is identified. JACPoL is accordingly introduced as a descriptive, scalable, and expressive policy language in JSON. JACPoL by design provides a flexible and fine-grained ABAC style (attribute-based access control) while it can be easily tailored to express other access control models. The design and implementation of JACPoL are illustrated together with its evaluation in comparison with other existing policy languages. The result shows that JACPoL can be as expressive as existing ones but more simple, scalable, and efficient. The performance evaluation shows that JACPoL requires much less processing time and memory space than XACML.

INTRODUCTION

Policies represent sets of properties of information processing systems (Clarkson&Schneider, 2010). Their implementation mainly rests on the IETF architecture (Yavatkar et al., 2000) initially introduced to manage QoS policies in networks. It consists in two main entities namely the PDP (Policy Decision Point) and the PEP (Policy Enforcement Point). The first one which is the smart part of the architecture acts as a controller the goal of which consists in handling and interpreting policy events, and deciding in

DOI: 10.4018/978-1-5225-8446-9.ch004

accordance with the policy currently applicable, what action should be taken. The decision is transmitted to the PEP which has to concretely carry it out.

Access control policies are a specific kind of security policies aiming to control the actions that principals can perform on resources by permitting their access only to the authorized ones. Typically, the access requests are intercepted and analyzed by the PEP, which then transfers the request details to the PDP for evaluation and authorization decision (Yavatkar et al., 2000). In most implementations, the stateless nature of PEP enables its ease of scale. However, the PDP has to consult the right policy set and apply the rules therein to reach a decision for each request and thus is often the performance bottleneck of policy-based access control systems. Therefore, a policy language determining how policies are expressed and evaluated is important and has a direct influence on the performance of the PDP.

Especially, in nowadays, protecting private resources in real-time has evolved into a rigid demand in domains such as home automation, smart cities, health care services and intelligent transportation systems, etc., where the environments are characterized by heterogeneous, distributed computing systems exchanging enormous volumes of time-critical data with varying levels of access control in a dynamic network. An access control policy language for these environments needs to be very well-structured, expressive but lightweight and easily extensible (Borders et al., 2005).

In this work, the authors investigate the relationship between the performance of the PDP, the language that is used to encode the policies and the access requests that it decides upon, and identify a set of key requirements for a policy language to guarantee the performance of the PDP. The authors argue that JSON would be more efficient and suitable than other alternatives (XML, etc.) as a policy data format in critical environments. According to these observations, the authors propose a simple but expressive access control policy language (JACPoL) based on JSON. A PoC (Proof of Concept) has been conducted through the implementation of JACPoL in a policy engine operated in reTHINK testbed (reTHINK Project Testbed, 2016)). At last JACPoL is carefully positioned in comparison with existing policy languages.

The main contribution of this work is therefore the definition of JACPoL, which utilizes JSON to encode a novel access control policy specification language with well-defined syntax and semantics. The authors identify key requirements and technical trends for future policy languages. They incidentally propose the new notion of Implicit Logic Operators (ILO), which can greatly reduce the size and complexity of a policy set while providing fine-grained access control. Quantitative evaluations of JACPoL by comparison to XACML show that JACPoL systematically requires much less time and memory space than XACML.The authors also elaborate on the applicability of JACPoL on ABAC model, RBAC model and their combinations or their by-products. Last but not least, their implementation leads to a novel and performant policy engine adopting the PDP/PEP architecture (Yavatkar et al., 2000) and JACPoL policy language based on Node.js[1] and Redis[2].

The remainder of this chapter is structured as follows. In Section 2, the problematic is refined by delimiting precisely its perimeter. Section 3 provides an illustration in depth with representative policy examples of the design of the policy language in terms of the constructs, semantics and other important features like Implicit Logic Operators, combining algorithms and implementation. Section 4 further qualitatively evaluates JACPoL and compares it with other existing access control policy specification languages. In section 5 a detailed performance evaluation is given. The ABAC-native nature of JACPoL is detailed in section 6 along with a comprehensive discussion on other possible application of JACPoL to ARBAC (Attribute-centric RBAC) and RABAC (Role-centric ABAC) security models. In Section 7 the authors summarize their work and discuss future research directions.