Chapter 12
Ontology-Based Authorization Model for XML Data in Distributed Systems

Amit Jain
University of South Carolina, USA

Csilla Farkas
University of South Carolina, USA

ABSTRACT

This research work proposes a Semantic-Aware Authorization Framework, called SAAF, for applying syntax independent authorization on eXtensible Markup Language (XML) documents. Our model supports secure data sharing in an open environment without the need for a centralized authority and supports application flexibility. We propose the use of data and application semantics, expressed as Resource Description Framework (RDF) ontologies, to specify security requirements for XML documents. XML documents are associated with their semantics (RDF ontologies) via mappings. The authors use these mappings and the corresponding RDF authorizations models to generate access control permissions for the mapped XML documents. The SAAF ensures the preservation of authorization permissions on XML data even if the syntax and the structure of the data are changed. Their method also aids the detection and removal of inconsistent authorizations on structurally different but semantically similar XML data.

INTRODUCTION

The rapid increase in the number of intelligent and autonomous technologies to support Internet usage created the need to represent web data and application semantics in a machine understandable way. Web data, used by humans and automated tools, exist in heterogeneous format in a distributed and open environment. Frequently, data and application semantics are embedded in the syntax and structure of the data. While such indirect representation of semantics is usually understandable for humans, it is not the case for automated tools. Moreover, security policies that are expressed over a specific representation of the data may not be applicable if the syntax or the

DOI: 10.4018/978-1-4666-2136-7.ch012

structure is modified. Web Services (WS), Service Oriented Architecture (SOA), and the Semantic Web are the state-of-the-art technologies supporting this distributed and open data and application paradigm. Ontologies are the building blocks of these technologies, providing a methodology to represent domain information and semantics in a machine understandable way. Using ontologies, syntactic data representations (such as the eXtensible Markup Language (XML), stream data, or unstructured data) can be associated with the corresponding data semantics. This enables the software applications and autonomous agents to understand and process the data intelligently without any human intervention or the need to hard code application specific semantics. WS are distributed Web applications, interacting with each other over the internet. They form a crucial component of SOA. WS operate and interact according to a set of published standards. These standards provide a way of developing decoupled software modules. Then the applications can share and process data among themselves irrespective of the heterogeneity of used languages, platforms or technologies. Current trends of Web applications indicate that WS will become a fundamental technology for Web-based applications. Web Services use XML as the basic format for data exchange. To provide security for WS applications, industry and standards committees, such as W3C and OASIS, have developed a set of standards. Security standards for XML formatted data are a fundamental component of these standards. Most of the XML security standards, however, use the syntactic and structural aspects of the XML. For example, XML access control models apply authorizations on the XML data syntax and fail to focus on the data and application semantics embedded in the syntax and structure. This can cause any change in XML format to deem the original security details invalid.

Let's consider a data sharing scenario between a Health Care Provider and an Insurance company in a Health Care domain illustrated in Figure 1. Both parties keep information about the patients in their own databases and exchange XML structured documents using WS. Here XML_1 is the message created by the Health care provider with the information pulled from its database. XML_2 represents the structure used by the Insurance Company to store the data for the shared nodes. The two XML trees have the same data but they differ in their syntax and structure. Currently authorization policies for these documents would be defined based on their syntax and structure. An access control policy for a patient data may require that a subject John is not permitted to read the illness information of the patients. Expressing this requirement would use different XPath expressions in the two XML documents. For example, element Patient/MedicalData/Illness in XML_1 may be represented as Patient/Data/HealthRecords/Diagnosis in the XML structure created by the Insurance Company (XML_2). The Web Service (WS_2) receiving the XML_1 would not know the semantics of the data and it may store and disseminate the information as XML_2. Hence the incoming XML data document is restructured and stored conforming to the target schema. This may results in accidental or intentional downgrade of for the data node "Illness" (i.e., changed from -ve to +ve access or from TopSecret to Public). In the case of a complex business workflow using WS, similar situations may arise. Furthermore, the current access control model will not be satisfactory for dynamic Web Service compositions. These compositions typically create business partners on the fly. Also periodically any of the partners may make changes to their respective storage schema for the shared data. This may also lead to inconsistent security policies that are hard to detect. As the enterprises are moving toward producing and sharing a large amount of data, leakage of sensitive data may occur very frequently. The developers of the policies must agree and incorporate data semantics to apply consistent security policies over different structured versions. Currently no automatic verification can be performed to verify