Classification-based Access Control Methodology in RDF

Abhishek P, Jayateerth S Deshpande and Kavi Mahesh

Centre for Knowledge Analytics and Ontological Engineering (KAnOE),
PES University, Bangalore 560085 India

drkavimahesh@gmail.com
http://www.kanoe.org

+91 9845290073

Keywords: Semantic Web, Resource Description Framework, Access control, Linked-Open-Data, Security.

Topics: Digital Preservation and Access Management, Data Schemes and Interoperability Formats (RDF), Web 3.0

Abstract:

Security is paramount to the continued development of any technology. Access control methodologies have practical impacts on the adoption of any technology in the current world. This article proposes an access control methodology for Resource Description Framework (RDF), which lies at the heart of semantic web. This methodology consists of two minimalistic techniques used for establishment and implementation of access control. Policies for access control can be represented in RDF with ease and flexibility .The techniques leverage the power of RDF and its sibling query language SPARQL. Potential applications include novel access control methodology for usage in RDF and NoSQL data stores including distributed data stores.

1. Introduction

Semantic Web is a vision of the future where the interoperability of devices and systems is seamless, a future where knowledge is boundless and evolves every day, where everyone can get the best of information in the simplest ways possible (Berners-Lee 2001). The possibilities that the semantic web envisions are becoming realities day by day. With the advent of Big Data, Cloud Computing and Internet of Things, the possibilities are less dreams and more reality than ever before. One of the technologies integral to the realisation of Semantic Web is RDF Resource Description Framework (RDF).

RDF has been evolving over the years since its inception towards providing 'interoperability between applications that exchange machine-understandable information on the Web' (W3C 1999). The capabilities of RDF have expanded over the years. Serialisation of RDF into XML facilitates and increases the ease of use. RDF makes up the Data Interchange layer of the Semantic Web.

Security is paramount for development of any technology in this day and age. Security Standards for the semantic web have also been discussed over the years (Thuraisingham 2005). Usage of RDF for policy specification and enforcement has also been proposed (Carminati 2004)(Jain 2006). Annotation-based, high-level access specification language has also been presented (Flouris 2010). This paper proposes an elegant and minimal technique of establishing access control using the classification characteristics of RDF.

2. Design

This access control can be established in three different ways.

1. Identity-based Access – Subject classification
2. Property-based Access – Predicate classification
3. Value – based Access – Object classification

Each of the three different methodologies above arises because of the triple style representation of facts intrinsic to RDF. Although the elicitation of the access control policy may vary for the three methodologies, the implementation of the policy is same i.e. query manipulation (refactoring).

2.1 Identity-based Access

As mentioned previously the Identity-based Access Control is based on the classification of subjects. We refer to it as Identity-based Access, since in a triple(fact), the subject is the identity to which the fact is most relevant. All subjects in RDF are Universal Resource Identifiers (URIs) (Berners-Lee 1994). Elicitation of the policy consists of adding triples classifying the subject into specific access class. The policy triple will contain the subject as its own subject classifying it as a member of a specific Access Class.

Consider the following triple associating an employee and his phone number.

<http://pes.edu/employee/JohnSmith><http://pes.edu/property/phonenumber>"9481955955".

This triple and the knowledge in it should be accessible only by a selective group of people, mainly his(John Smith) superiors. Consider an access class , LeaderAccess to which the above triple has to be associated. This can be done by adding a simple triple as given below.

<http://pes.edu/employee/JohnSmith>rdf:member<http://pes.edu/classes/LeaderAccess>

The definition of the LeaderAccess class and associating it with the respective managers will be done with a few but not numerous triples.

2.2 Property-Based Access

Property-based Access is based on the classification of predicates of a triple. The predicate of a triple provides the context and meaning, i.e. the relationship between the subject and the object. Property may help define the subject, or relate one or more entities. Predicate classification hence provides us a means to classify the triple (or a set of triples) based on its meaning. Similar to Identity-based Access, a predicate can be classified into an access class which can then be mapped to roles.

Consider following triple associating an employee to his salary.

<http://pes.edu/employee/JohnSmith><http://pes.edu/property/salary>"40000".

This triple associates an employee and his salary. Similar triples may exist for each of the employees. By classifying this predicate into an access class, effectively one classifies all similar triples into a single class.

<http://pes.edu/property/salary>rdf:member <http://pes.edu/classes/FinanceAccess>.

The above triple established the classification and hence the associated access control.

This methodology can be utilised where numerous triples of similar meaning exist in the triple store. The definition of the access class is similar to that mentioned previously.

2.3 Value-based Access

Value-based Access is based on the classification of objects in the triples. Objects can either be URIs or literals (strings numbers). Classifying an object which is an URI is similar to the methodology mentioned in the subsections 2.1 and 2.2 . Classifying an object which is a literal necessitates one more step.

Value-based Access can be used where, although the existence of such a triple is known, the value of the object should not be. Consider that every employee is reviewed by their team leaders at the end of each calendar year. This score, although the employee knows exists, should not be accessible by him. Classifying the object signifying the value can be classified into a specific access class, restricting the access to it.

Consider the following example where the object of the triple is a URI.

<http://pes.edu/employee/JohnSmith><http://pes.edu/property/nextPosting><http://pes.edu/location/Bangalore>.

This object of the triple is the next location he will be posted to, which should only be known to select people.

The classification triple will be as follows,

<http://pes.edu/location/Bangalore>rdf:member<http://pes.edu/classes/ValueAccess/nextPosting>.

Classification of this URI will not be identical to any of the other two methodologies since, this classification is only asked for when the object of the triple is URI and not subject.

2.4 Marking the classification:

This way of implementing access control requires some way to represent whether an access control classification is Identity-based, Property-based or Value-based. This can be achieved by using specialised access classes for each of the methodologies.

3. Implementation

The mode of implementing the different methodologies mentioned in the previous section is the same, namely, using regular-expression based query manipulation. This entails identification of the access controls of the user and appending of the query body with condition triples which lead to inferences of the access control policies

and hence the results are access controlled. The conditional triples are formed based on the identity of the user who is querying the store.

Consider an example, where an employee searches for phone numbers of all people in the data store. The query would be as follows

```
SELECT  ?a ?c where
{
        ?a  <http://pes.edu/property/phonenumber> ?c.
};
```

Here the result would consist of all phone numbers in the store user/entity names and their phone numbers. If an Identity-based Access control is used such that he can only see the phone numbers of members of his/her own department, the resulting manipulated query would be as below

```
SELECT  ?a ?c where
{
        ?a  <http://pes.edu/property/phonenumber> ?c.
        ?a rdf:member <http://pes.edu/classes/Department1>.
};
```

For Property-based Access, the conditional triple is a triple classifying the predicate into the access class as shown below.

```
SELECT  ?a ?c where
{
        ?a  <http://pes.edu/property/salary> ?c.
};
```

is changed to

```
SELECT  ?a ?c where
{
?a  <http://pes.edu/property/salary> ?c.
        <http://pes.edu/property/salary> rdf:member <http://pes.edu/classes/Finance>
};
```

In Value-based Access, the conditional triple is the triple classifying the object to the access class, regardless of it being a literal or a n URI, but the literal has to be associated with its URI counterpart.

```
SELECT ?a ?b ?c where
```

```
{
        ?a <http://pes.edu/property/nextPosting>  ?c.
};
```

is changed to

```
SELECT ?a ?b ?c where
{
        ?a <http://pes.edu/property/nextPosting>  ?c.
        ?c rdf:member <http://pes.edu/classes/ValueAcess/nextPosting>.
};
```

4. Conclusion and Future Work

The work presented in this paper provides an elegant method to implement access control in a minimalistic manner. It is work in progress. Complex applications would require development of more versatile policies. Creation of such policies will necessitate development of methods to seamlessly combine the three different methodologies and more flexible ways of developing access policies in RDF. Usage of these policies with other forms of data storage and representation such as NoSQL databases has to be explored. Combining such representation with a query processor will also provide a more fundamental way of providing access control in a query-based store. Nevertheless, the elegant idea illustrated in this paper shows how semantic web technologies can be used to define and manage classification-based access control to critical data.

References

1. Tim Berners-Lee, J Hendler, O Lassila. 2001.
   The Semantic Web
   Scientific American, May Issue

2. World-wide Web Consortium. 1999
   Resource Description Framework (RDF) Model and Syntax Specification
   REC-rdf-syntax-19990222

3. B Thuraisingham, 2005
   Security Standards for the Semantic Web
   Computer Standards & Interfaces, Volume 27 Issue 3 : 257-268

4. B. Carminati, E. Ferrari and B. Thuraisingham, 2004
   Using RDF for policy specification and enforcement.
   Database and Expert Systems Applications,2004
   Proceedings. 15th International Workshop on, 2004, pp. 163-167.

5. A Jain, C Farkas, 2006

Secure resource description framework: an access control model,

Proceeding, SACMAT'06 Proceeding of the ACM Symposium on Access control models and technologies, pp. 121-129

6. G Flouris, I Fundulaki, M Michou, G Antoniou, 2010

Controlling access to RDF graphs.

Future Internet - FIS 2010, Berlin, Germany,September 20-22, 2010. Proceedings pp. 107-117

7. Tim Berners-Lee, 1994

Universal Resource Identifiers in WWW,

Request for Comments: 1630

CERN