Using Semantic Web Technologies to Specify Constraints of RBAC

Wu Di, Lin Jian, Dong Yabo, Zhu Miaoliang
College of Computer Science, Zhejiang University
{wudi, apolin, dongyb, zhum}@zju.edu.cn

Abstract

Role-based access control (RBAC) models have generated a great interest in the security community as a powerful and generalized approach to security management. One of important aspects in RBAC is constraints that constrain what components in RBAC are allowed to do. There are lots of research have been achieved to specify constraints for secure system developers. However more work is need urgently to met requirements for interoperability of machine and people understandable constraints specification in open and distributed environment. In this paper we propose another approach to specify constraints using Semantic Web technologies. The Web Ontology Language (OWL) specification of basic RBAC components and constraints are described in detail.

Keywords: RBAC, constraint, Semantic Web, OWL

1. Introduction

Role-based access control (RBAC) models have generated great interest in the security community as a powerful and generalized approach to security management [1]. In RBAC, users are assigned to roles and roles are associated with permissions. The permission determines what operations a user assigned to a role can perform on information resources. Instead of specifying all the accesses each individual user is allowed, access authorizations on objects are specified for roles. RBAC has ability to model organizational structure and their potential to reduce administrative overheads. Authorization constraint is a fundamental aspect of RBAC. Although the importance of constraints in RBAC has been recognized for a long time, insufficient research has been devoted to constraints. Especially an interoperable specification of constraints is needed that is easy to understand and use, and also allows for the analysis in open and distributed environment.

In this paper, we show how the Semantic Web (SW) technologies [2] can be used to specify RBAC constraints. The Semantic Web is an extension of the current web in which better defines the meaning of the information, enabling computers and people to work better in cooperation. The W3C is designing OWL (Ontology Web Language) [3] a semantic markup language, which provides formalized knowledge expression and more flexibility, sharing a great deal of common semantics about expressing access control constraints.

The rest of the paper is organized as follows. In Section 2, some related work about RBAC constraints is introduced. With the SW technologies the basic definitions of RBAC model are proposed in Section 3 and Section 4 particularly describes the RBAC constraints. Last a conclusion is made and prospects of the future work are looked forward.

2. Related Work

Tidswell and Jaeger [4] propose an approach to visualizing access control constraints. They point out the need for visualizing constraints and the limitations of previous work on expressing constraints. A drawback of their work is that they created a new notation for specifying constraints and it is not clear how the new notation can be integrated with other widely-used design notations. The approach described in this paper utilizes a standardized technology.

A large volume of research exists in the area of specification of access control policies. Formal logic-based approaches [5] are often used to specify security policies. They assume a strong mathematical background which makes them difficult to use and understand. Other researchers have used high-level languages to specify policies [6]. Although high-level languages are easier to understand than formal logic-based approaches, they are not analyzable. Semantic Web technologies, on the other hand, provide an easy-to-use approach supported by mechanisms for detecting problems with the specifications.

Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'05)
0-7695-2405-2/05 $20.00 © 2005 IEEE

3. RBAC Specification

In this section we describe RBAC in terms of OWL. Figure1 describes some basic components of RBAC. Because the constraints of RBAC is the core problem discussed in this paper, only basic component in RBAC will be involved with the requirement of constraints specification.

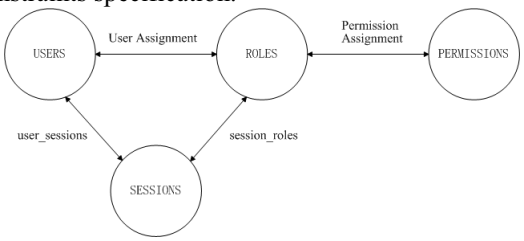

Figure 1 The basic components of RBAC

- **Definition of User, Role, Permission and Session**

The basic definitions of User, Role, Permission and Session are similar. The basic definition with OWL is as follows:

```
<owl:Class rdf:ID="User"/>
<owl:Class rdf:ID="Role"/>
<owl:Class rdf:ID="Permission"/>
<owl:Class rdf:ID=”Session”/>
```

- **Definition of relation among User, Role and Permission**

There are two types of many-to-many relation among User, Role and Permission.

User to role assignment relation:

```
<owl:ObjectProperty rdf:ID=" hasRole ">
   <rdfs:range rdf:resource="#Role"/>
   <rdfs:domain rdf:resource="#User"/>
 </owl:ObjectProperty>
```

Permission to role assignment relation:

```
<owl:ObjectProperty rdf:ID="hasPermission">
   <rdfs:range rdf:resource="#Permission"/>
   <rdfs:domain rdf:resource="#Role"/>
</owl:ObjectProperty>
```

- **Definition of relation among User, Role and Session**

A function mapping each session to the single user is expressed:

```
<owl:FunctionalProperty rdf:ID="belongTo">
   <rdfs:range rdf:resource="#User"/>
   <rdfs:domain rdf:resource="#Session"/>
</owl:FunctionalProperty>
```

A function mapping each session to a set of roles can be expressed as follows:

```
<owl:ObjectProperty rdf:ID="hasActiveRole">
   <rdfs:domain rdf:resource="#Session"/>
   <rdfs:range rdf:resource="#Role"/>
</owl:ObjectProperty>
```

4. RBAC Constraints Specification

Constraints are important aspect of RBAC and are often regarded as one of the principal motivations behind RBAC. Due to the limitation of length of the paper, only Separation of Duty Constraints and Prerequisite Constraints based on specification of RBAC are described in detail.

4.1. Separation of Duty Constraints

Separation of duty is a well-known principle for preventing fraud by identifying conflicting roles. Separation of duty may be enforced either statically or dynamically.

Static Separation of Duty (SSD) constraints aim to prevent conflict of interests that arise when user gains permissions associated with conflicting roles (roles that cannot be assigned to the same user). SSD constraints place constraint on the assignment of users to roles, that is, membership in one role that takes part in an SSD constraint prevents the user from being a member of the other conflicting role. In other words, a user can have at most one of mutually exclusive roles.

The definition of conflict relation between roles can be expressed:

```
<owl:ObjectProperty rdf:ID="conflictRole">
   <rdfs:range rdf:resource="# Role "/>
   <rdfs:domain rdf:resource="#Role"/>
</owl:ObjectProperty>
```

Then the SSD constraint applied to role can be specified using OWL DL expression as follows:

$$hasRole(?user,?role1) \wedge conflictRole(?role1,?role2) \rightarrow \neg hasRole(?user,?role2)$$

This conflicting notion can be applied to other elements such as user and permission in RBAC. The definition of conflict relation between users or permissions is similar to definition of role conflict relation.

The SSD constraint applied to permission means that a user can have, at most, one conflicting permission acquired through roles assigned to the user. This type of constraints prevents mistakes in role-permission assignment. The following OWL DL expression ensures that two conflicting permissions cannot be assigned to the same role:

$$hasPermission(?role,?permission1) \wedge conflictPermission(?permission1,?permission2) \rightarrow \neg hasRole(?user,?role2)$$

The SSD constraint applied to user should be also considered. This type of constraints places constraint on the users that can be assigned to roles. Assignment of a user that takes part in a constraint to a role

prevents the other conflicting user being assigned to the role. The OWL expression is as follows:

$$hasRole(?user1, ?role) \wedge conflictUser(?user1, ?user2) \rightarrow \neg hasRole(?user2, ?role)$$

Dynamic Separation of Duty (DSD) constraints aim to prevent conflict of interests as well. DSD constraints place restrictions on the roles that can be activated within the same user session. Suppose that a user has the supervisor roles and inherits permissions from both accounts payable manager role and purchasing manager role. It may be acceptable for the user not to activate these two conflicting roles at the same time. The DSD constraint is expressed using the OWL as follows:

$$hasActiveRole(?session, ?role1) \wedge conflictRole(?role1, ?role2) \rightarrow \neg hasActiveRole(?session, ?role2)$$

4.2. Prerequisite Constraints

The concept of prerequisite roles is based on competency and appropriateness, whereby a user can be assigned to role only if the user already is assigned to role's prerequisites. For example, only users who are already assigned to the project role can be assigned to the testing role in that project.

The definition of prerequisite relation between roles can be expressed:

```
<owl:ObjectProperty rdf:ID="prerequisiteRole">
   <rdfs:domain rdf:resource="# Role "/>
   <rdfs:range rdf:resource="#Role"/>
</owl:ObjectProperty>
```

Prerequisite Role constraints are expressed using OWL expressions as follows:

$$hasRole(?user, ?role1) \wedge prerequisiteRole(?role2, ?role1) \rightarrow hasRole(?user, ?role2)$$

Prerequisite constraints can be also applied to permission. For example, in UNIX operation systems permission to read a file requires permission to read the directory in which the file is located.

5. Conclusion

The access control in open, heterogeneous and distributed systems poses important challenges to support interoperability of access control information. This paper has described a new approach to use Semantic Web technologies to specify the RBAC constraints. Separation of duty constraints, prerequisite constraints and cardinality constraints has been specified with OWL. These representations can be validated by ontology-related reasoning tools (such as RACER, JENA, etc.) when designing and analyzing RBAC systems. In conclusion, the semantic approach of RBAC constraints is the foundation to achieve semantic interoperability among the different components of access control systems.

The work described in this paper focuses on specifying the static description of RBAC constraints. It does not provide a systematic modeling approach that can be used to create applications with RBAC constraints and validate them. In our future work, we will focus on using Semantic Web technologies to specify a general family of RBAC models in RBAC96 [7]. The implementation of the systematic modeling approach will also be accomplished.

References

[1] J. Bacon, K. Moody, and W. Yao, "A Model of OASIS Role-Based Access Control and Its Support for Active Security," *ACM Trans. Information and System Security*, vol. 5, no. 4, Nov. 2002.

[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", Scientific American may 2001.

[3] Mike Dean, Guus Schreiber, "OWL Web Ontology Language: Reference", *World Wide Web Consortium*, 18 August (2003), http://www.w3.org/TR/2003/CR-owl-ref-20030818/

[4] J.E. Tidswell and T. Jaeger, "An Access Control Model for Simplifying Constraint Expression", *In Proceedings of the 7th ACM conference on Computer and communications security*, Athens, Greece, November 2000, pp. 154-163.

[5] S. Barker, "Security Policy Specification in Logic", *In Proceedings of the International Conference on Artificial Intelligence*, Las Vegas, NV, 2000, pp. 143-148.

[6] M. Hitchens and V. Varadarajan, "Tower: A Language for Role-Based Access Control", *In Proceedings of the Policy Workshop*, Bristol, U.K., 2001.

[7] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "Role-based access control models" *IEEE Computer*, Volume 29, Number2, February 1996, pp. 38-47.