AN EXTENDED ROLE-BASED ACCESS CONTROL FOR WEB SERVICES

Yi-qun Zhu, Jian-hua Li, Quan-hai Zhang
Department of Electronic Engineering, Shanghai JiaoTong Univ.,Shanghai 200030, China

Keywords: Access Control, Web services, Web services security, Authorization.

Abstract: A key challenge in Web services security is the design of effective access control schemes that can adequately satisfy Web services security requirements. Despite the recent advances in Web based access control, there remain issues that impede the development of effective access control models for Web services environments. One of them is the lacks of dynamic role management and attributes access control for Web services. In this paper, we present a dynamic attribute-based role-based access control model (DARBAC) to address the issues. The proposed approach introduces authorization group, which is used to dynamically manages roles and privileges, and attribute based access control mechanism which is used to protect the services and services parameters. We outline the configuration mechanism needed to apply our model to the Web services environments.

1 INTRODUCTION

Web services are well known and widely accepted because of features of reusing and interoperability. Web services are loosely coupled applications using well-known XML protocols for representation and communication across the Internet. Compared to centralized systems and client-server environments, Web services are much more dynamic and distributed, and their security poses unique challenges. One of the challenges is how to design effective dynamic access control model in XML-based Web services.

Several access control models have been proposed to protect the resources, such as DAC (Discretionary Access Control), MAC(Mandatory Access Control), RBAC(Role Based Access Control) (Sandhu,1996) (Ferraiolo,2001). These models are designed to protect the resources in static environments and LAN-based applications, which make them not really meet security needs of the complicated Web services environments.

The access control is already becoming the hot topic in the field of Web services security. Both the subjects that request Web services and objects that provide the resources are active and dynamic entities. Some models that consider dynamic feature of subject and object are proposed. The paper(Yan Han,2000) introduces the notion of role-playing to implement the dynamic management of activated roles and proposes a general role-based object access control model. Based on the task-based access control(Deng JB,2003), the paper(Xu Feng,2005) proposes a service-oriented access control model that combines the dynamic object of role-playing(Yan Han,2000) and designs its security architecture(Xu Feng,2004) A solution of the dynamic feature of role(Xu Feng,2005) is provided, and traditional protected objects are replaced by services. However, both services and their parameters must be protected from unauthorized users in global services(Roosdiana,2004). The model(Xu Feng,2005) can not provided a solution for protecting both services and their parameters. The paper(Roosdiana,2004) proposes a model that can protect both services and parameters. However, it can not take the dynamic feature of role into account and can not completely express dynamic features of role.

In this paper an extended role based access control model is presented to address above problems. By introducing “authorization group” and “attribute access control” to the basic RBAC model, we can address dynamic features of role and protect the services attributes that consist of services and parameters. The remainder of the paper is organized as follows. The second section presents a dynamic attribute-based RBAC model (DARBAC model) for Web services and discusses the related issues. The section 3 concludes our paper.

2 DARBAC MODEL

2.1 The Elements

Figure 1 depicts elements of DARBAC model and their relationships with each other. The DARBAC model introduces authorization group to dynamically manage privileges in Web services. It protects attribute of web service by enforcing attribute access control at two levels, service level and service attribute level. Service level access control ensures that only authorized users invoke the web services. Attribute level access control is applied only when the authorized users pass the service authorization.

DARBAC model is composed by the below entity set and the relations. The following definition formalized the DARBAC model:

(1) U, R, P, and S (users, roles, permissions, and services respectively).

(2) Service attributes (SA): Service attributes are objects that are used as parameters or returned values for Web services.For one service get_price<title,project>,the attribute consists of service name get_price and service parameters title and project.In the process of invoking services, the required information is passed in to web services and its operation results are passed out of web services by using the service attributes. The service attributes for a service s are represented SA(s), and $SA(s) \subseteq SA$.

(3) Access mode: An access mode is a type of access to service attributes, such as read, write, modify, etc. For example, a role r is granted read on param1,which is used by service s. So, the user who assigned to r can read param1 used by s as its parameters or returned values for Web services. $param1 \in SA(s)$. A role that be able to execute Web services must be assigned permissions for the service and access mode for service attributes. An access mode can be created by combining one or more access modes. For example,modify can be the access mode that combining read and write.

(4) UA,$UA \subseteq U \times R$, a many to many user-to-role assignment relation. User assignment is defined as a tuple <user, role>. UA(r) denotes the set of users that assigned to role r.

(5) RR,$RR \subseteq R \times R$, a role-to-role relation. A partial order on R is role hierarchy. Roles can be created and ordered in a hierarchical manner. A role can contain other roles in a role hierarchy. When a role contains other roles, it inherits all of the permissions to the services and service attributes that the other roles have. When a role r_1 contains other roles, such as r_2 and r_3, the role r_1 can inherit all the permissions of role r_2 and r_3 for the services and service attributes. If role r_2 and r_3 hold conflicting access modes on the same service attributes, the role r_1 has all the access modes on the service attributes.

(6) Service:A service is an abstraction of the operations provided by the system on its objects, and a service is designated by the service name s. Services use the parameters as the carrier of information that be passed in to(or out of) the services. A service assigns a minimum permission to each of its parameters.

(7) PA, $PA \subseteq P \times R$, a many to many permission-to-role assignment relation. For a role that be able to execute service and get the information, it must be granted permissions to service's attribute.

(8) SPA, $SPA \subseteq SP \times R$, a many to many service-permission-to-role assignment relation. When a role is able to execute a service, it must be granted permission to the service. SPA is defined as a tuple <roles, services>.

(9) SAPA, $SAPA \subseteq SAP \times R$, a many to many service-attribute-permission-to-role assignment relation. SAPA is defined as a tuple <roles, service attributes>. A role must be granted permissions to both the service and service's

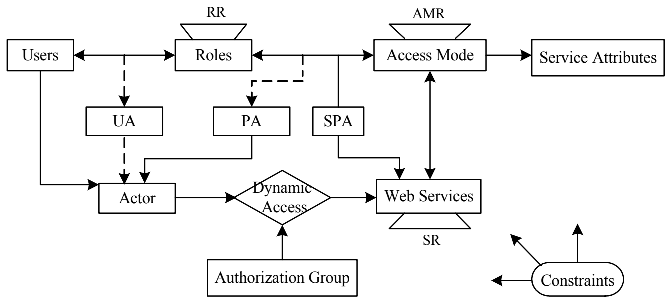

Figure 1:The Elements of DARBAC and its Relations with Elements in the Model.

attribute to be able to execute service. After a role passes the service authorization, it can get the service information by using access mode of service attribute in the SAPA.

(10) Actor: Actor is a dynamic object. Every actor can be defined as a triple <user, role , t>, where t is the period time of the actor. When a user u that is be assigned a role r activates r, system creates an actor a, which is <u, r, t>. The constraints of r are passed to the actor a accordingly.

(11) Authorization group: Authorization group is one process of authorization. Figure 2 depicts elements and relations involved in authorization group. One authorization group consists of a actor, permissions(P), and its lifecycle. Authorization group can be defined as a triple<actor, permissions, lifecycle>. When user u activates role r that is assigned to u, system creates an actor a that is <u,r,t>. So the actor a has permissions of role r , which are represented P(r). P(r) consists of permissions to the service and service attribute. In the process of implementing authorization group, the permissions of the actor are called enabled permissions(EP). EP consist of enabled services(ES) and enabled service attributes(ESA).

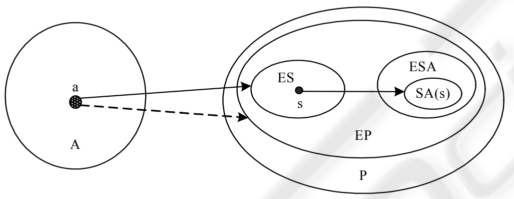

Figure 2: Authorization Group.

2.2 Dynamic Management

Authorization group has five basic states, which are dormant, invoked, valid, invalid, hold. Authorization group transform its state according to the dynamic system information. Figure 3 depicts state transition of authorization group.For example, when system is busy, authorization group transform from valid to hold and suspended. After completing authorization, authorization group turns into invalid state.

The lifecycle of authorization group, which is designated by the variable Lifecycle_AG, is decided by services providers according to the system state. For example, there are large numbers of users to request the Web services, so that the services provider can not undertake the busy condition. The Web services providers can adjust the system condition by adusting the size of Lifecycle_AG. If AG is invalid state, the actor can not access to the objects. When AG is in valid state, the role of corresponding actor can access to the authorization objects, service and service attributes. At the same time, t of the actor is the time between the current time and the time that actor is created. If t is in scope of Lifecycle_AG, actor is in the work. When t exceeds the value of Lifecycle_AG, actor is inactivated and is destroyed. As a result, the role can not execute the service and attributes.

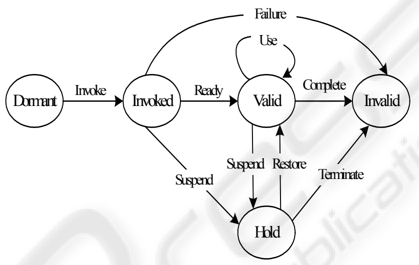

Figure 3: State Transition of Authorization Group.

2.3 Permission Assignment

DARBAC model enforces attribute access control at two levels, service level and service attribute level. If user u wants to get information of service s, u must be assigned one or more roles. Role r, that is assigned to the user u and can execute service s, must be granted permissions to both service s and attributes of s. Once role r passes the service level access control, it also must pass the service attribute access control. Roles maintain two lists, one is the list of services that roles that are assigned to the users have permissions to execute, the other is the list of access modes of attributes that roles have permissions to access. For a role that be able to execute service and get the information, it must be granted permissions to both the service and service attributes.

Services use the attributes as the carrier of information that be passed in to(or out of) the services. A service assigns a minimum permission to each of its parameters in order to enforce service attribute level access control. That is to say, a service must at least have read access to its input parameter, write access to its output parameters.

2.4 Authorization

This section discusses how a user who nominates a role is granted permissions to get the requested service. Authorization changes with the context of web services environments. In the DARBAC model, the access control process is partitioned into four phases, which are user assignment, permission assignment, role activate, dynamically adjust privilege. The decision to grant permissions to invoke a service is based on the following :

(1) DARBAC model enforces service and service attribute access control. When a user requests a service, it must be assigned at least one role. Role r, that is assigned to the user u and can execute service s, must be granted permissions both to service s and to attributes of s. Once role r passes the service control, it also must pass the service attribute control.

(2) A user is the services requestor. A user is given a unique identity within a services provider.When the user u requests the service s, system passes the user's identity authentication and checks whether the user u has assigned the role r that can execute the service. After user u activates the role r that is assigned to the user u, system creates the dynamic actor a to execute the permissions of service and service attributes for the user u.

When user u invokes the service, system checks whether the authorization group(AG) is in valid state. If AG is in invalid state, the role r of the actor can not work. If the checking result is positive, the corresponding actor begins to work and check for value of t in the actor. If t is in scope of Lifecycle_AG, actor is in the work. When t exceeds the value of Lifecycle_AG, corresponding actor also is destroyed. As a result, the corresponding role and the permissions of service and service attribute are revoked, and authorization also is stopped.

3 CONCLUSION

In this paper, we present a DARBAC model for access control for Web services. It not only supports dynamic access control based on security environments condition, but also protects both services and service attributes. By introducing the notion of authorization group and attribute access control mechanism, the model can enforce the dynamic privileges management for Web services. Compared to traditional RBAC, the DARBAC model can manage privileges dynamically, and suites global service features.

REFERENCES

Ravi S.Sandhu, Edward J.Coyne. *Role-Based Access Control Models*, IEEE Computer, 1996,29(2):38-47.

Ferraiolo D.F., Sandhu R., Guirila S., Kuhn D.R., Chandramouli R.. *Proposed NIST Standard for Role-Based Access Control*,ACM Transactions on Information and System Security, Pages 224–274,Vol. 4, No. 3, August 2001.

Yan Han,Zhang Hong,Xu Man-Wu.*Object Modeling and Implementation of Access Control Based on Role*,Chinese Journal of Computers, 2000,Vol.23 No.10.

Deng JB,Hong F. *Task-Based Access Control*, Journal of Software,2003,14(1):76-82.

Xu Feng,Lin Guoyuan,Huang Hao,Xie Li. *Role-based Access Control System for Web Services*, In:Proceedings of the Fourth International Conference on Computer and Information Technology (CIT'04),Wuhan,2004,357-362.

Xu Feng, Lai Hai-Guang, Huang Hao, Xie Li. *Service-Oriented Role-Based Access Control*, Chinese Journal of Computers,2005,Vol.28 No.4.

Roosdiana Wonohoesodo, Zahir Tari. *A Role based Access Control for Web Services*, In:Proceedings of the 2004 IEEE International Conference on Services Computing (SCC'04), 2004.