ISBN 978-981-11-0008-6
Proceedings of 2016 6th International Workshop on Computer Science and Engineering (WCSE 2016)
Tokyo, Japan, 17-19 June, 2016, pp. 477-480
doi: 10.18178/wcse.2016.06.080

Approach to Implementing Authentication Service with Role-based Access Control

Poowanart Korbkum and Yachai Limpiyakorn [+]

Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand

Abstract. Authentication is typically required to prevent unauthorized users. In addition to satisfy the security dimension, restricted permission grants are also in organization concerns for web services access. In order to effectively serve a wide range of users and enable accesses to various services by non-specified domains, the traditional one-to-one authentication service is not adequate. This paper thus presents an approach to implementing the authentication service with the notion of role-based access control. The implementation of immediate authentication service including the parts of API and Administration web service is presented to demonstrate the flexible usage of web services on different domains.

Keywords: role-based access control, permission, authentication, web services.

1. Introduction

For system security, user authentication is required when accessing web services. Today authentication services are available in a variety of formats. The general authentication service is one-to-one service, i.e. it requires the new user to register once prior to accessing the web site. Nowadays, the popular authentication services model is SSO (Single Sign-on) that allows the user to sign in to multiple websites with an existing account. For instance, SSO is used as the authentication protocol for Facebook and Twitter.

However, Authentication is merely the mechanism to prevent the unauthorized access to the system. Another function unit is still required to determine permission grants for authorized users to access certain web services or information on the services.

The traditional access control of web services is designed based on user permission and user roles to access certain features provided on web services. When new users have registered to web services, the administrator or user manager assigns the user permission or user role to allow the new user for accessing particular feature area. Many web services have an automatic system or authentication system to assign the user permission after user registration. However, the engine does not flexible enough for the web service which has many service types and frequently serves new service or task for different user groups.

In literature, Won and Seog [1] introduced a method based on the concept of role-based access control to realize the access control of the multiple web services in distributed web server and to control document view depending on the user's role by using memory cookie web browser. Li and Wu [2] presented a simple design and implementation of the authorization system based on RBAC. The database of authorization system was developed consisting of 3 data tables: 1) user table contains user data, 2) department table stores all department services that users can access, and 3) user's right table stores all access permission of each user based on the user type. In this paper, the development of RBAC [3] authentication service is presented to manage a wide range of users and provide the flexible usage of web services on different domains.

[+] Corresponding author. Tel.: + 668 2218 6959; fax: +668 2218 6955.
E-mail address: Yachai.L@chula.ac.th.

2. Role-based Access Control (RBAC)

Role-based access control [3] is very effective and more advantages than traditional access control. The notion of RBAC is that permissions are associated with roles, and users are assigned appropriate roles. User and permission are brought together with a role and included into a user group [4].

The core RBAC model [5] is shown in Fig. 1. The model includes a set of basic data elements: users (USERS), roles (ROLES), objects (OBJECTS), operations (OPERATIONS), and permission (PERMISSIONS). Users are assigned roles, which in turn, are associated with permissions. Permissions indicate types of operations on certain objects. Objects can be web pages, forms, menus etc. The model also includes a set of sessions (SESSIONS) where a session is a mapping between a user and the current subset of roles that is assigned to the user.

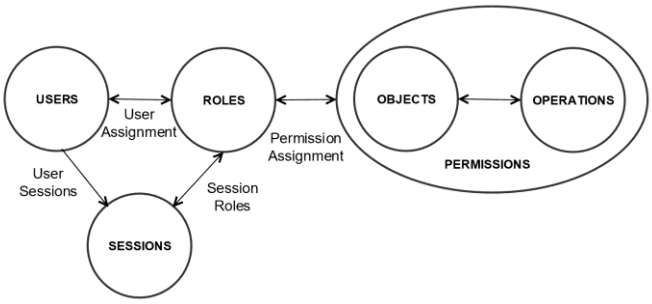

Fig. 1: Core RBAC Model [5].

3. Implementation

Based on core RBAC model shown in Fig. 1, the authentication service has been implemented. As shown in Fig. 2, the Web Service provides a means of user login. The request of user name and password will be sent to the Authentication Service for the permission grant returned to the Web Service via the Application Programing Interface (API) of the Authorization Service. For the new user, registration is required. The new user is then assigned a role which determines the permission for particular operations on certain feature areas.

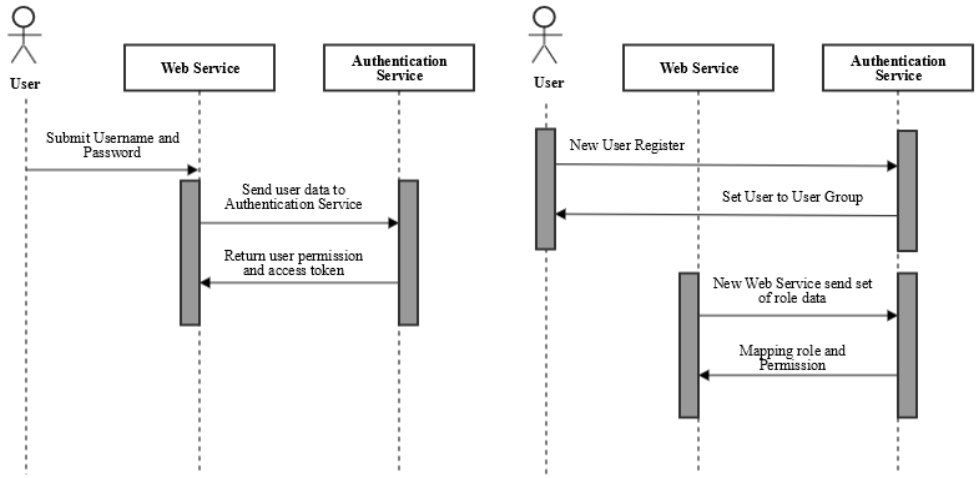

Fig. 2: Sequence Diagram of Proposed RBAC Authentication Service.

The two major components of the Authentication Service are developed: API, and Web Administration. The API interacts with the Web Service for responding the request of user authentication and granting the access permission to the user indicated by the user's role. A role denotes a set of permissions that allows the user to access a set of predefined tasks.

The Web Administration is responsible for user management, role management, and task management as illustrated in Fig. 3. The diagram reflects the relationship between the user group and role that are joined into the user group role. A user group is a cluster of users sharing the same task usages. A user can be assigned as a member of several user groups. Multiple user groups can share the same role, called a user group role. A

user group can be a member of multiple user group roles.

The role management agent is responsible for determination of roles for a web service when it is created. A new role will be established to grant the permission of tasks allowed for related user groups.

When a new user is created, the user will be categorized into appropriate user groups so that the user will be granted the permissions associated with the proper roles. In case there is none of user group suitable for the new user, the new user group will be created and it will be added to the related user group roles. The new user will be added to the newly created user group afterwards.

A role is a set of permissions that users can access to the predefined tasks. The task management agent determines the policy of each task or defines the roles relevant to an individual task.

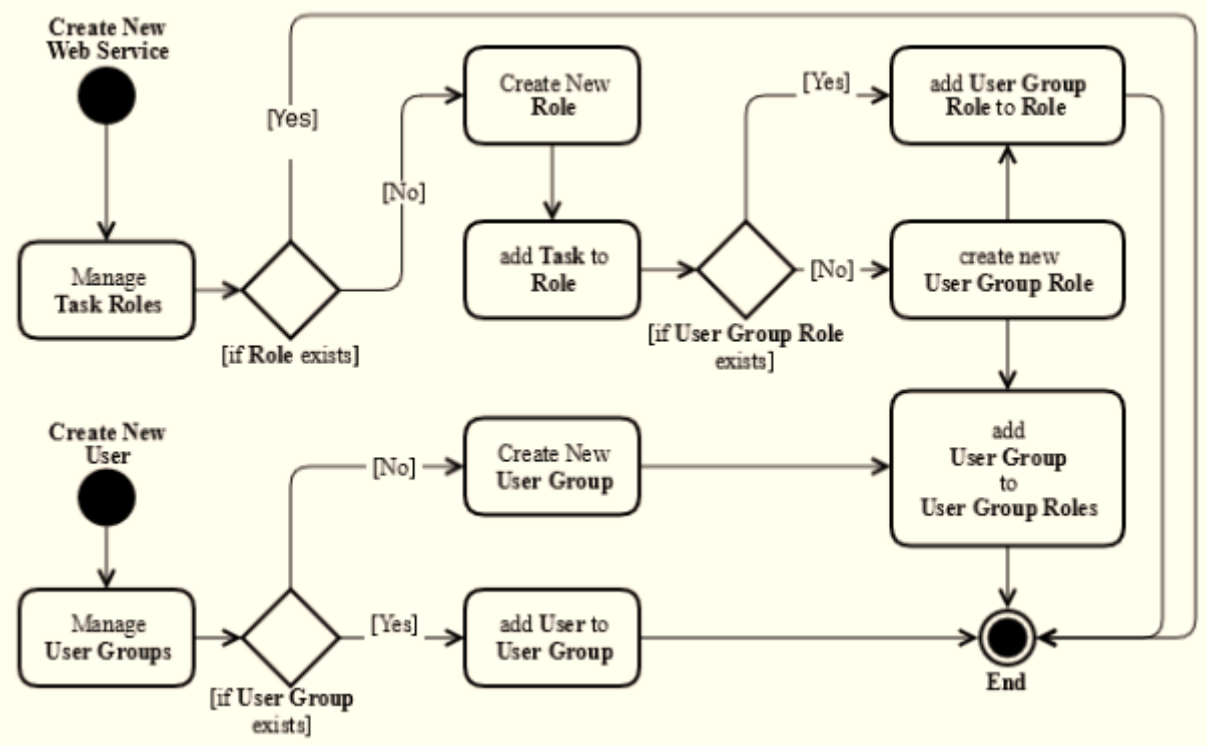

Fig. 3: Activity diagram of web administration process.

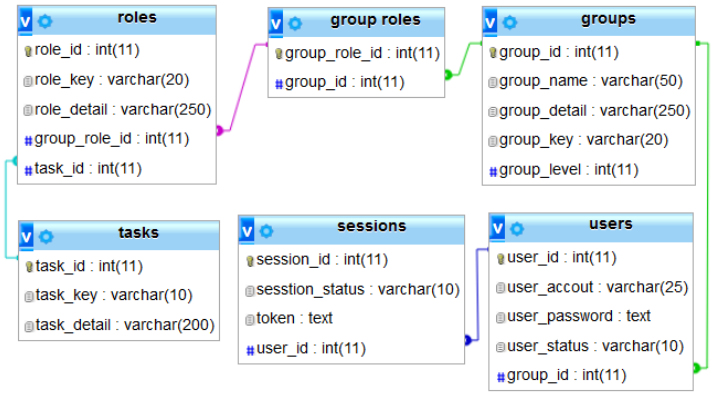

Fig. 4: Excerpt database schema connected to authentication service.

Fig. 4 shows parts of database schema designed for the Authentication service. The tasks table contains all permission grants for all existing tasks. The roles table stores all roles associated with a set of tasks referencing task_id in the tasks table and a set of user group roles referencing group_role_id in the group roles table. The group roles table contains a set of user groups associated with the same role. The groups table contains user categories and relations between a role and users. The attribute of group_level indicates group priority, i.e. the groups with higher level acquire the permissions of lower level groups. The users table

contains user information such as account and password. The sessions table stores user authentication status. When logging in, an access token is created and referenced with user_id to respond to the web service that requests for user authentication.

Fig. 5 demonstrates the communication between the Web Service and Authentication Service. When a user logs in, the Web Service will send the user data to Authentication Service for verification via server side script. If the user account and password are correct, the Authentication Service will create an access token or update the existing token, and then look for the user permission from the roles table in database. The access token together with the user permission will be returned to the Web Service in JSON format.

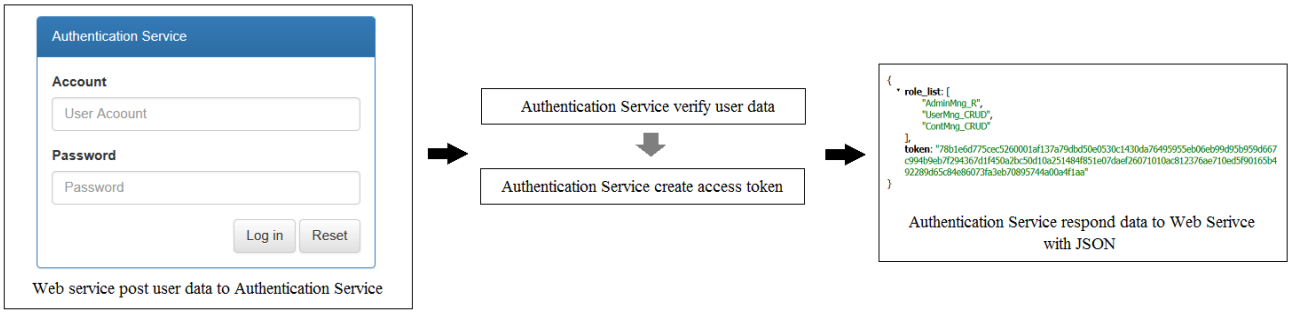

Fig. 5: Example communication between web service and authentication service.

4. Conclusion

In this paper, an authentication service is implemented based on the core RBAC model. The main idea is that the permission or access control is based on roles. A user can be assigned multiple roles. A user and permissions are brought together with a role and included into a user group. The presented approach would help manage the secure access to organization web services with a wide range of users as well as effectively provide the permission grants for various user groups. This would lessen the burden of web administrator when a web service has many users and user types. Rather than wasting the efforts to assign users to the new web service one by one each time, the comprehensive roles will be identified for the web service. The permissions associated with a role will then be dispersed to all the users residing the user group that maps to the role defined by User Group Roles automatically.

5. References

[1] B. S. Won, and P. Seog. Implementing Web Access Control System for the Multiple Web Servers in the Same Domain Using RBAC Concept. Eighth International Conference on Parallel and Distributed Systems. Oct. 2001, pp. 768-773.

[2] F. Li, and H. Wu. Design and Implementation of Authorization System Based on RBAC. 7th International Conference on Intelligent Human-Machine Systems and Cybernetics. Aug. 2015, pp. 502-504.

[3] International Committee for Information Technology Standards. Role Based Access Control. American Nation Standards for Information Technology. Feb. 2004, pp. 11-19.

[4] D. F. Ferraiolo, and D. Richard Kuhn. Role-Based Access Controls. 15th National Computer Security Conference. Oct. 1992, pp. 554-563.

[5] R. S. Sandhu, E. J. Coyne, H. L. Feistein, and C. E. Youman. Role-Based Access Control Models. IEEE Computer, Vol 29. Feb. 1996, pp. 38-47.