J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

A Role-Based Access Control System API Supporting External Authority Interface☆

Jin Ma[1] Hyunah Kim[2] Minjae Park[3*]

ABSTRACT

In industries that are operating various enterprise systems, new systems are integrated and operated in accordance with each period. In particular, when a new system is to be integrated, one of the major considerations is the single sign-on part for integrating and operating the authentication. To implement this authority system using role-based access control method, an extension method for access control method is needed. Therefore, in this paper, we design an extended role-based access control model for interworking with legacy authority system and provide its APIs. The extended role-based access control model is a model in which external authority information, which holds authority information in the authority information, is added. And we describe operations that the REST Web APIs are based on these models. In this paper, the method is described in the back-end APIs and can be implemented as an operation of an extended role-based access control system based on the method.

☞ keyword : Role-Based Access Control, External Interface, Web API, Restful API

1. Introduction

Traditionally, the applicability of role-based access control model to commercial systems is apparent from its widespread use[1]. Baldwin [2] describes a database system using roles to control access. Nash and Poland [3] discuss applying role based access control concept to cryptographic authentication devices commonly used in the banking industry. In fact, we would like to apply this concept for manufacturing industry area that had the legacy authority system.

Generally, in order to apply new authority system, the legacy authority system should be replaced with the new authority system. In other words, the existing system should disappear and it should be integrated into the new system. However, in many cases, the user still desires to use legacy system, too. Therefore, we extend role-based access control model and implement role based access control interface that supports legacy authority. It has occasionally introduced a new system.

2. Related Work

The basic concept of the role-based access control model[5, 6] has long been proposed and utilized. And there have also been studies about the standard[7-9]. Also there have been various studies on design and implement[10-17] about functionalities. application fields and etc.

Especially, [13, 15] have an in-depth approach to the model in terms of the model itself and formal specification. [14, 16] describe role-based access control features in systems commercial like database Management systems and MLS systems. Among them, [10, 17] are significant in that it implies the possibility of expanding the model variously. Because we are also interested in implementing the system, [12] it is meaningful in that it deals with aspects of implementation. We also conducted a new study on the extension and an implementation method in this study.

[1] Dept. of Super Computing, KISTI, Daejun, 34141, South Korea
[2] Dept. of Computer Science, Kyonggi University, Suwon, 16227, South Korea
[3] Dept. of Computer Software, Daelim University, Anyang, 13916, South Korea
* Corresponding author (mjpark@daelim.ac.kr)

[Received 29 August 2017, Reviewed 6 September 2017(R2 28 November 2017, R3 11 January 2018), Accepted 21 February 2018]

☆ A preliminary version of this paper was presented at ICONI 2016 and was selected as an outstanding paper.

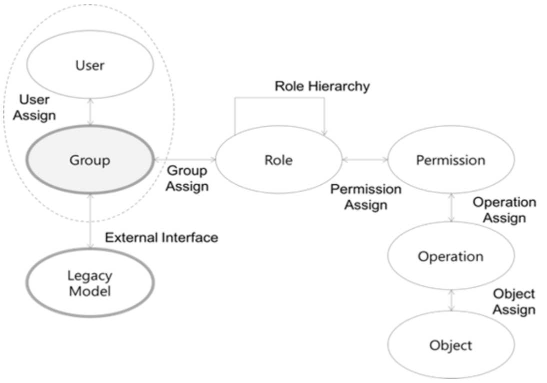
☆ This research was supported by the EDISON Program through the National Research Foundation of Korea(NRF) (No. NRF-2011-0020576), and the KISTI Program (No. K-18-L12-C06-S01)

http://dx.doi.org/10.7472/jksii.2018.19.2.27

3. An Extended Role-Based Access Control Model

The basic role-based access control model[4] is extended with the group concept that has external or internal type. When defining an extended role-based access control model, the following conventions are useful.

- S = Subject(User) = A person agent
- G = Group = A group agent
- R = Role = Job function or title which defines an authority level
- P = Permissions = An approval of a mode of access to a resource
- SE = Session = A mapping involving S, G, R and/or P
- SA = Subject Assignment
- GA = Group Assignment
- PA = Permission Assignment
- RH = Partially ordered Role Hierarchy. RH can also be written: ≥ (The notation: x ≥ y means that x inherits the permissions of y.)
  - A subject can have a group.
  - A group can have multiple roles.
  - A role can have multiple subjects.
  - A role can have many permissions.
  - A permission can be assigned to many roles.
  - An operation can be assigned to many permissions.



(Figure 1) An extended role-based access control model for external authority interface

  - A permission can be assigned to many operations.

The extended role-based access control model is able to integrate with external model or the system like the legacy model. It is represented in Figure 1, which is depicted above consisting of objects and relationships.

Main Objects mean subject, group, role and permission and main relationships mean user assign, group assign, and permission assign.

4. Back-end System Operations for User Authentication

In order to apply an extended role-based access control model to the system, we describe how to run on the back-end API. In fact, fetching information from the extended role-based access control model information is a matter of resources management. However, because it relates to authentication, it handles not only resources management, but also authentication management. Considering such things, we can see that it can be operated as shown figure 2.

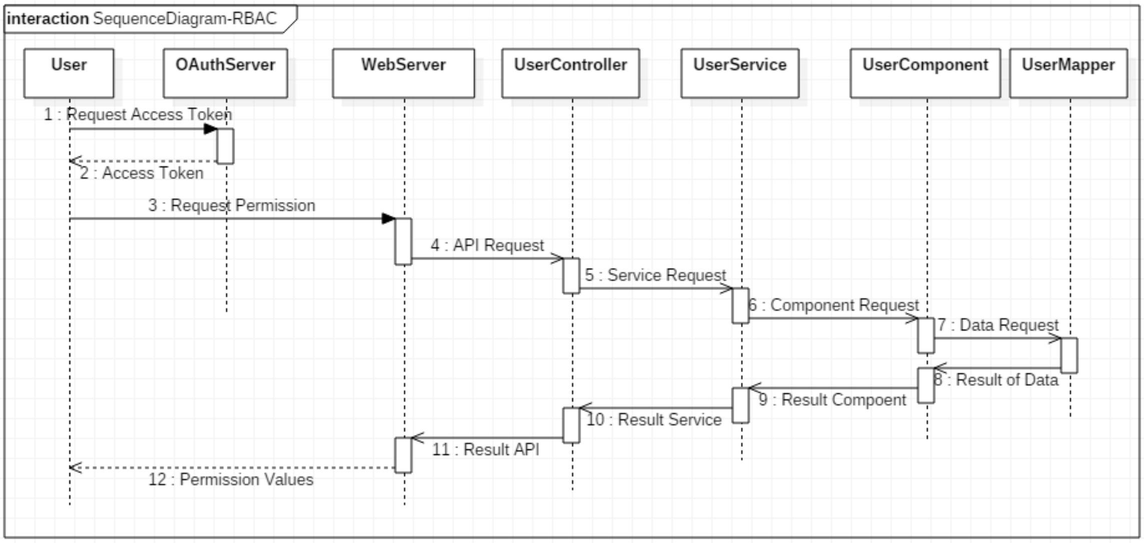
A user receives the authentication value that is an access token through OAuthServer by requesting access token. After that, the user requests the authority from the server. From this point, the user can request the permission of extended role based access control model and retrieve the value.

In the case of OAuth authentication, it will fetch a value with extended role-based access control model authority considering external system, so it will be possible to authenticate by external system which includes value for access to external system.

Designing Web APIs based on this information are described in Chapter 4. It is based on restful interface and is operated through separate methods for users, roles and permissions.

5. A Role-Based Access Control System Web APIs for External Authority Interface

We have described an extended role-based access control system with functions that consists of users, roles, permissions



(Figure 2) Sequence diagram of system applied the extended role-based access control model

and groups. The relationship between users, roles, permissions and groups follows the basic role-based access control model.

The process that the users with id and password log in is the same as a normal system, but the access to token value, which is the information to be acquired after log-in, contains values for access to the related system that is internal or external.

Table 1, 2, 3, 4, 5, and 6 represent the result of each GET method by implementation. Especially, in Table 2, 3, and 4, the groups method has attributes groupId, description, type,

(Table 1) GET method for '/rbac/users' API

```
GET: /rbac/users
[
  {
    "userId": "MJPARK",
    "groupId": "ADMIN_GROUP",
    "name": "mjpark",
    "password": "****",
    "description": null,
    "email": null,
    "phone": null,
    "birthday": null,
    "sex": null,
    "position": null,
    "department": null,
    "companyPhone": null,
    "type": null
  }
]
```

(Table 2) GET method for '/rbac/groups' API

```
GET: /rbac/groups
[
  {
    "groupId": "ADMIN_GROUP",
    "description": "enable duplicated",
    "type": "I",
    "role": [
      {
        "condition": "All",
        "roleId": "ADMIN"
      },
      {
        "condition": "All",
        "roleId": "P_ADMIN"
      },
      {
        "condition": "All",
        "roleId": "E_ADMIN"
      }
    ]
},
{
  "groupId": "EXTERNAL_GROUP",
  "description": "enable duplicated",
  "type": "E",
  "role": [
    {
      "condition": "All",
      "roleId": "ADMIN"
    }
  ]
}
]
```

and multi-roles. The attribute ‘type’ has a value ‘I’ or ‘E’. ‘I’ means internal and ‘E’ means external interface to authority system.

(Table 3) GET method for ‘/rbac/groups/{groupId}’ API has Internal Type

```
GET: /rbac/groups/ADMIN_GROUP

{
  "groupId": "ADMIN_GROUP",
  "description": "enable duplicated",
  "type": "I",
  "role": [
    {
      "condition": "All",
      "roleId": "ADMIN"
    },
    {
      "condition": "All",
      "roleId": "E_ADMIN"
    },
    {
      "condition": "All",
      "roleId": "P_ADMIN"
    }
  ]
}
```

(Table 4) GET method for ‘/rbac/groups/ {groupId}’ API has External Type

```
GET: /rbac/groups/EXTERNAL_GROUP

{
  "groupId": "EXTERNAL_GROUP",
  "description": "enable duplicated",
  "type": "E",
  "role": [
    {
      "condition": "All",
      "roleId": "ADMIN"
    }
  ]
}
```

(Table 5) GET method for ‘/rbac/roles’ API

```
GET: /rbac/roles

[
  {
    "roleId": "ADMIN",
    "description": null,
    "childRole": [
      {
        "roleId": "E_ADMIN",
        "description": null,
        "childRole": [
          {
            "roleId": "POWER_USER",
            "description": null,
            "childRole": []
          }
        ]
      },
      {
        "roleId": "P_ADMIN",
        "description": null,
        "childRole": [
          {
            "roleId": "POWER_USER",
            "description": null,
            "childRole": []
          }
        ]
      },
      {
        "roleId": "POWER_USER",
        "description": null,
        "childRole": []
      }
    ]
  },
  {
    "roleId": "POWER_USER",
    "description": null,
    "childRole": []
  },
  {
    "roleId": "P_ADMIN",
    "description": null,
    "childRole": [
      {
        "roleId": "POWER_USER",
        "description": null,
        "childRole": []
      }
    ]
  },
  {
    "roleId": "E_ADMIN",
    "description": null,
    "childRole": [
      {
        "roleId": "POWER_USER",
        "description": null,
        "childRole": []
      }
    ]
  }
]
```

(Table 6) GET method for ‘/rbac/roles/{roleId}’ API

```
GET: /rbac/roles/ADMIN

[
  {
    "roleId": "ADMIN",
    "description": null,
    "childRole": [
      "E_ADMIN",
      "P_ADMIN",
      "POWER_USER"
    ],
    "permission": [
      "1",
      "7",
      "12"
    ]
  }
]
```

6. Conclusions

In this paper, we have proposed a new role-based access control model that is able to be integrated with legacy authority model. And we describe a method for back-end operations and how to implement its operating APIs. It implements back-end API. Besides it can be easily combined front-end service as restful API.

This is significant for role-based access control model in that it is not a new attempt in the function extension itself, but it is a meaningful extension of new functionality of internetworking with the legacy system.

We expect that it will be operated as service in real industry that aims to introduce a new system covering the legacy system

Reference

[1] Atluri, Vijayalakshmi and David F. Ferraiolo. “Role-Based Access Control.” Encyclopedia of Cryptography and Security (2011). http://doi.org/10.1007/978-1-4419-5906-5_829

[2] R.W. Baldwin, “Naming and Grouping Privileges to Simplify Security Management in Large Databases,” In IEEE Symposium on Computer Security and Privacy, 1990. http://doi.org/10.1109/RISP.1990.63844

[3] K.R. Poland M.J. Nash, “Some Conundrums Concerning Separation of Duty,” In IEEE Symposium on Computer Security and Privacy, 1990. http://doi.org/10.1109/RISP.1990.63851

[4] https://en.wikipedia.org/wiki/Role-based_access_control

[5] D.F. Ferraiolo and D.R. Kuhn (1992) "Role Based Access Control" 15th National Computer Security Conference, Oct 13-16, 1992, pp. 554-563. - introduced formal model for role based access control.

[6] R. S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. "Role-Based Access Control Models", IEEE Computer 29(2): 38-47, IEEE Press, 1996.- proposed a framework for RBAC models. http://doi.org/10.1109/2.485845

[7] R. Sandhu, D.F. Ferraiolo, D, R. Kuhn (2000), "The NIST Model for Role Based Access Control: Toward a Unified Standard," Proceedings, 5th ACM Workshop on Role Based Access Control, July 26-27, 2000, Berlin, pp.47-63 - first public draft of the NIST RBAC model and proposal for an RBAC standard. http://doi.org/10.1145/344287.344301

[8] D.F. Ferraiolo, R. Kuhn, R. Sandhu (2007), "RBAC Standard Rationale: comments on a Critique of the ANSI Standard on Role Based Access Control", IEEE Security & Privacy, vol. 5, no. 6 (Nov/Dec 2007), pp. 51-53 - explains decisions made in developing RBAC standard.

[9] D.R. Kuhn, E.J. Coyne, T.R. Weil, "Adding Attributes to Role Based Access Control", IEEE Computer, vol. 43, no. 6 (June, 2010), pp. 79-81. http://doi.org/10.1109/MC.2010.155

[10] Hwang Yu-Dong, Park Dong-Gue, "Extended GTRBAC Delegation Model for Access Control Enforcement in Enterprise Environments", Journal of Internet Computing and Services, Vol. 7 No.1, 2006.2, 17-30.

[11] Seng-phil Hong, Hyun-me Jang, "Applied Method of Privacy Information Protection Mechanism", Journal of Internet Computing and Services, Vol. 9, No. 2, 2008.4, 51-59

[12] Kyung-Soo Joo, Jung-Woong Woo, "An Object-Oriented Analysis and Design Methodology for Security of Web Applications", Journal of Internet Computing and Services, Vol.14, No.4, 2013.8, 35-42

[13] D.R. Kuhn, "Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role-Based Access Control Systems" Second ACM Workshop on Role-Based Access Control. 1997 http://doi.org/10.1145/266741.266749

[14] R. Chandramouli, R. Sandhu, "Role Based Access Control Features in Commercial Database Management Systems," 21st National Information Systems Security Conference, October 6-9, 1998

[15] S. Gavrila, J. Barkley, "Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management" (1998), Third ACM Workshop on Role-Based Access Control. http://doi.org/10.1145/286884.286902

[16] D.R. Kuhn. "Role Based Access Control on MLS Systems Without Kernel Changes" Third ACM Workshop on Role Based Access Control, October 22-23,1998 http://doi.org/10.1145/286884.286890

[17] R. Sandhu, D. Ferraiolo, R. Kuhn, "The NIST Model for Role Based Access Control: Towards a Unified Standard," Proceedings, 5th ACM Workshop on Role Based Access Control, July 26-27, 2000, Berlin, pp.47-63 http://doi.org/10.1145/344287.344301

◐ 저자소개 ◑

마　진 (Jin Ma)
2010년 광운대학교 컴퓨터소프트웨어학과 졸업(학사)
2012년 광운대학교 대학원 컴퓨터과학과 졸업(석사)
2012년～2015년 (주)비스텔 선임연구원
2015년～현재 한국과학기술정보연구원(KISTI) 계산과학공학센터 연구원
관심분야 : 데이터 통합, 빅 데이터, 분석시스템, 분산컴퓨팅, 정보검색
E-mail : majin@kisti.re.kr

김 현 아 (Hyunah Kim)
2001년 나사렛대학교 전산정보학과(이학사)
2003년 경기대학교 일반대학원 전자계산학과(이학석사)
2009년 경기대학교 일반대학원 전자계산학과(이학박사)
2007년～2011년 경기도지역협력연구센터산하콘텐츠융합소프트웨어연구센터 (GRRC)
2012년～2018년 경인여자대학교 초빙교수
2018년～현재 경기대학교 융합교양대학 조교수
관심분야 : 프로세스 기반 정보시스템, 워크플로우, 프로세스 기반 e-Learning, e-Learning 서비스
E-mail : hyuna486@kgu.ac.kr

박 민 재 (Minjae Park)
2004년 경기대학교 전자계산학과(이학사)
2006년 경기대학교 일반대학원 전자계산학과(이학석사)
2009년 경기대학교 일반대학원 전자계산학과(이학박사)
2009년～2017년 ㈜비스텔 선임/책임/수석연구원
2017년～현재 대림대학교 컴퓨터소프트웨어과 조교수
관심분야 : 프로세스 기반 정보시스템, 워크플로우, 시설물 관리 소프트웨어, 소프트웨어 플랫폼, IoT
E-mail : mjpark@daelim.ac.kr