An Application of Directory Service Markup Language (DSML) for Role-based Access Control (RBAC)

Dongwan Shin & Gail-Joon Ahn

Lab. of Information Integration, Security and Privacy (LIISP)
Dept. of Software and Information Systems
UNC Charlotte
Charlotte, NC 28223

Joon S. Park

School of Information Studies
4-291 Center for Science & Technology
Syracuse University
Syracuse, NY 13244-4100

Abstract

Directory service markup language (DSML) prescribes how to manipulate directory services information in XML, and thus it facilitates sharing of directory information as XML fragments among XML-based applications. In this paper, we describe how to leverage DSML for role-based access control on XML-based web applications which often need collaboration within or beyond a single enterprise boundary. Compared with previous works in this area, we show that our approach can solve the problems of a previous LDAP-oriented solution. We discuss the security architecture based upon server-pull model and its components. We also demonstrate the feasibility of our approach through a proof-of-concept implementation. Finally, several issues from our experience are discussed as well.

Keywords: *Access Control, Role-based, Directory Service, Directory Service Markup Language (DSML)*

1. INTRODUCTION

The *Web* has been evolving continuously since its inception and its technology has as well. Web-based business solutions have replaced traditional business solutions for closed computer network, purporting to satisfy the needs of the parties involved in business activities. However, as web-based solutions and information they are dealing with increase in number, volume, and complexity, there are growing demands for an efficient and protected way of managing or sharing such information. For example a user's data, i.e., his/her profile may be used repeatedly in a chain of such solutions for proper operations. Interoperability of such data improves efficiency, negating extra communication for the data, and security mechanisms prevent abuse, alteration, and loss of the user's data. The extensible markup language (XML) has played a key role in shaping current web-based solutions trends. XML offers easier manipulation of data, providing for both context and interoperability of data. As for security, on the other hand, it is likely that web-based solutions are vulnerable to security threats and their security is relatively hard to achieve.

To uphold security in web-based solutions is a multi-disciplinary task. It includes from authentication and access control on one side to intrusion detection on the other. Recently access control, which is the issue we address in this paper, has been recognized as the major requirement to secure enterprise resources, facilitating delegation of administration and personalized contents. Role-based access control (RBAC) has been acclaimed and proven to be a simple, flexible, and convenient way of managing access control [1][2]. In RBAC, access control depends upon the roles of which a user is a member, and permissions are assigned to the roles.

Proceedings of the 26 th Annual International Computer Software and Applications Conference (COMPSAC'02)
0730-3157/02 $17.00 © 2002 **IEEE**

This extremely simplifies management of permissions, eliminating complexities in managing ACLs. Also, RBAC can be easily reconfigured to comply with different organizational access control policies [3][4][5] [10].

Several researches have been carried out on implementing RBAC on the web [6][7][8][9]. In particular Park et al. investigated LDAP-oriented approach to implement RBAC on the web in the *server-pull* model [8]. They used an LDAP directory service server as a role server. The web server retrieves the role information in the role server for access control decision through LDAP over SSL. However, there are some issues to be addressed further regarding the role information; its interoperable usage in collaborating environment, bypassing firewalls to retrieve it, and its administration. For instance, to use LDAP, the LDAP client software must be installed and used in the client side (each web server in our case). Furthermore, if the LDAP communications go through a firewall, which usually sits in between different organizations, we need to configure the firewall to allow the LDAP communications (e.g., port 389). Under a strict security policy, the security administrator may not approve this request.

The objective of our work is to implement a role-based authorization service for web-based solutions using directory service markup language (DSML). DSML prescribes how to represent directory services in XML, and thus it offers the benefits of both XML and directory service. Directory service provides a viable solution for storing and manipulating enterprise-wise information in a scalable manner. We identify a set of components that is necessary to pursue our goal and develop an appropriate system architecture based upon *server-pull* model. Also, we demonstrate the feasibility of our architectures by providing the proof-of-concept prototype implementation using commercial off-the-shelf (COTS) technologies. Our implementation is based on RBAC model in [1] and DSML V2 [15].

This paper is organized as follows. Section 2 gives an overview of Directory Service Markup Language (DSML). Section 3 describes our approach to design a role-based authorization service using DSML and the system architecture. Section 4 discusses implementation details. Section 5 concludes the paper.

2. DIRECTORY SERVICE MARKUP LANGUAGE (DSML)

Directory is strongly believed to be an optimal solution for making enterprise resources available to different systems in an organization. It maps names to objects and also allows such objects to have attributes. Thus, objects can be looked up by either their name or attributes and their attributes can be also made available for further operations. For example, a user might be represented by a directory object that has as attributes the user's email address, role information, telephone number, and so on. The user can be searched based on either his/her name or attributes, and the user's role can be also retrieved and used for a certain functionality such as role-based access control.

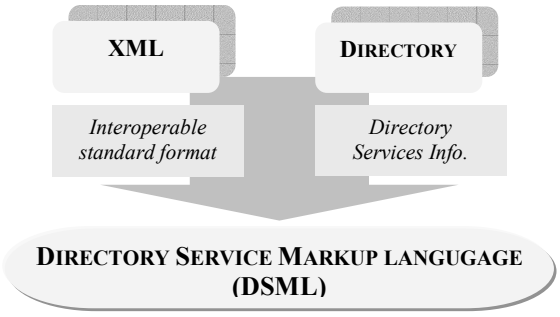

Figure 1 DSML as directory service in XML

As shown in Figure 1, DSML is based on XML, proposing to describe the structure and content of directory services information in an XML document. It is in essence an XML version of the familiar Lightweight Directory Access Protocol (LDAP). Using DSML, any XML-enabled application can look up, add, modify, and delete the directory services information and leverage the benefits of directory services such as the scalability, replication, and security [15]. DSML was originated from the need of interoperability of different vendors' directory service and the need of adoption of XML [13][14]. In 1999, it was

initiated by Bowstreet Software Inc. and under its leadership other major software companies, including Microsoft and Novel, joined DSML working group. DSML version 2 was recommended as OASIS Standard in 2001. Compared with the previous version, DSML version 2 supports the bindings such as the SOAP request and response binding, allowing for directories to be manipulated via XML (DSML version 1 represents only static contents of directory services). In comparison with LDAP, DSML supports a method of grouping operations to be expressed in a single request and allows accessing a directory through firewalls [15].

3. OPERATIONAL ARCHITECTURE

We propose an approach to provide a role-based authorization for web-based applications existing in collaborating environments in this section. Our approach leverages RBAC and DSML. DSML allows an efficient and interoperable usage of role information from directory services, and RBAC provides the framework for building a role-based environment in directories and enforcing role-based access control.

Figure 2 shows an operational architecture for our approach. Our architecture is designed to work seamlessly in both a single organization and multiple organizations. It consists of six components. They are client, web server, access control enforcer, DSML gateway server, role database, and role administration server. Using role administration server, security administrators initially set up role-based environments such as defining and creating roles, assigning users to roles, associating permissions to roles, and so on. Those role-based environments are stored in a role database.

Since our architecture is based upon the *server-pull* model, one of the two approaches identified in [9] in terms of obtaining attributes on the Web, the web server pulls a user's role information from the role database. A user needs to be authenticated before he/she requests resources. Authentication, however, does have little bearing upon our architecture because we want our architecture to be independent of authentication schemes, and we believe it provides our architecture with more flexibility. The communication between a client and a web server is on HTTP.

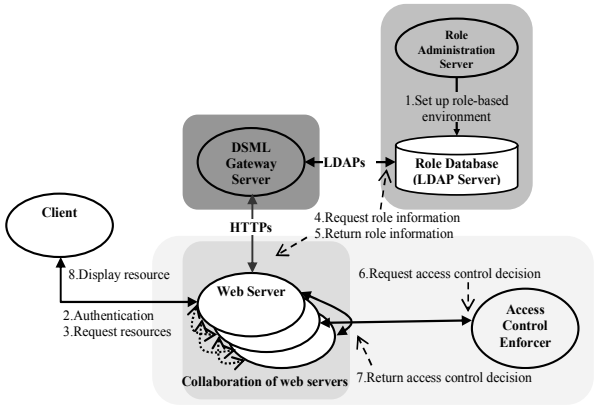

Figure 2 Operational architecture in server-pull model

After successful authentication the web server pulls the user's roles from the role database. In request and acquisition of the user's roles, DSML gateway server, capable of SOAP communication, sits in between the web server and the role database. HTTP is the communication protocol in between the web server and DSML gateway server, while LDAP is in between the DSML gateway server and role database. Both communications are encrypted for confidentiality and integrity of the communications using SSL (Secure Socket Layer). The web server sends the request for the user's role in XML format to DSML gateway server. DSML gateway server then parses the request, accesses the role database using LDAP protocol, and retrieves the user's role information. The role information is transformed into DSML and sent back to the web server. The communications between the Web servers and the role database through the DSML gateway server enable the interoperable RBAC services for different organizations.

In a simplest case a web server is enough to provide all the resources that the user requests. However, it seems to be true that the recent trends such as portalling and aggregation of web solutions make it infeasible for a single web server to satisfy all the services that the user wants.

Hence multiple web servers located within or beyond a single organization need to collaborate to offer what the user wants. The user's role information available to all the web servers is indispensable for them to enforce an efficient and coherent access control. In our architecture DSML fragment of the user's role credentials facilitate sharing of the credentials among collaborating web servers. Each web server sends the user's roles and resource identifiers to access control enforcer for authorization.

4. IMPLEMENTATION DETAILS

In our implementation, we make use of COTS products with minimal modifications to meet our objectives. Our implementation plan is simple such that it is based upon the access control decision procedures discussed in our operational architecture. A client could be any web browser that supports JavaScript, and we used Internet explorer 6.0 as a client. Using JavaScript, we performed the validity test of the data that a user might enter such as user ID and password (We used an authentication using user ID and password in our implementation). As for the web server, we used Apache web server with Java Servlet container and SSL-enabled. Java Servlet was employed in order to implement the web server's interaction and communication with the client as well as DSML gateway server. We used the public source code available from Userland Software Inc. and modified it in order to use as DSML gateway server. Its functionality is to accept remote procedure calling using HTTP as a transport and XML as an encoding and to make LDAP request to an LDAP server for the user's roles. The following DSML fragment is the result of requesting a user's role information from DSML gateway server. That fragment shows that the assigned role to Alice Taylor is **Power user**.

```
<dsml:dsml xmlns:dsml="http://www.dsml.org/DSML">
 <dsml:directory-entries>
   <dsml:entry dn="uid=alice, …">
            …
            …
   <dsml:attr name="cn">
     <dsml:value>Alice Taylor</dsml:value>
   </dsml:attr>
   <dsml:attr name="rbacRole">
     <dsml:value>Power user</dsml:value>
            …
            …
 </dsml:entry>
 </dsml:directory-entries>
</dsml:dsml>
```

We developed an access control enforcer with an engine for making access control decision as its major component. Web servers request access control decision by sending the requested objects (resources) identifier and the user's roles to the access control enforcer.

Role administration server was developed to help system administrators easily construct the components of role administration and orchestrate them to build a role-based infrastructure. Our role administration server is a Java-based stand-alone application. Java Naming and Directory Interface (JNDI) was used to establish the communication between the application and a role database. Netscape Directory Service 5.0 was used for the role database. Figure 3 shows the interface of role administration server when it is about to perform the operation of assigning users to roles.

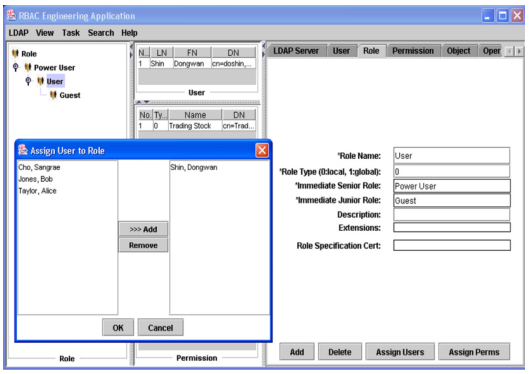

Figure 3 Role administration server

5. DISCUSSION AND CONCLUSION

In this paper we described the benefits of role-based access control and the features of DSML. Then, we proposed an approach to implement a role-based authorization service leveraging DSML. A security architecture was discussed on the basis of *server-pull model*. In addition, we demonstrated the feasibility of our approach

through a proof-of-concept implementation. Compared with previous works, our approach could ease the inconvenience in handling a user's role information resulted from LDAP-oriented approach, enabling its interoperable usage among collaborating web servers within or beyond a single organization boundary.

In addition to the architecture based upon *server-pull* model, our experience indicated that an architecture based on *user-pull* model is desirable under certain situations. For example, if a user is willing to cooperate in obtaining his/her roles and wants more control on the usage of his/her roles, the *user-pull* based architecture is preferable. In such an case the user's role information can be contained in a secure medium, for example, attribute certificate, which has been proposed and discussed in [11][12].

We admit that DSML has limited usages in authentication or authorization scheme by its working group. Security Assertion Markup Language (SAML) is an alternative XML format which is designed to be used extensively in authentication and authorization schemes [16]. SAML is an XML-based framework for exchanging security credentials such as role information. Those security credentials are expressed in the form of assertions about subjects. One of our future researches will be based on this observation.

Also, our work indicated that we need a systematic policy management which can manage different or conflicting policies from multiple organizations even though they are using same role-based framework. So as to handle this issue, we should consider policy coordination, policy translation and delegation between multiple organizations.

Last but not least, we have learned that we need to investigate more exhaustive methods for role administration. Well defined formal specifications are required in both permission assignment and in user assignment for establishing role-based infrastructure. Hence we believe such attempts could contribute to the standardization of role administration, and its standardization could help developers or administrators design, develop, or administrate role-based systems much easier.

REFERENCES

[1] R. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. "Role Based Access Control Models," *IEEE Computer 29 (2)*, February 1996.

[2] D. Ferraiolo, J. Cugini, and D.R Kuhn. "Role Based Access Control: Features and Motivations," In *Annual Computer Security Applications Conference*, IEEE Computer Society Press, 1995.

[3] G. Ahn and R. Sandhu. "Role-based Authorization Constraints Specification," *ACM Transactions on Information and System Security*, 3(4), November 2000.

[4] R. Sandhu. "Role-hierarchies and Constraints for lattice-based access control," In *Proceedings of 4th European Symposium on Research in Computer Security*, Rome, Italy, 1996.

[5] R. Sandhu, "Role-hierarchies and Constraints for lattice-based access controls," *Proceedings of 4th European Symposium on Research in Computer Security*, Rome, Italy, 1996.

[6] G. Ahn, R. Sandhu, M. Kang, and J. Park. "Injecting RBAC to secure a Web-based workflow system," In *Proceedings of 5th ACM Workshop on Role-Based Access Control*. Berlin, Germany, July 2000.

[7] J. Park and R. Sandhu. "RBAC on the Web by Smart Certificates," In *Proceedings of the 4th ACM Workshop on Role-Based Access Control*, Fairfax, VA, October 1999.

[8] J. Park, G. Ahn, and R. Sandhu. "RBAC on the Web using LDAP," In *Proceedings of the 15th IFIP WG 11.3 Working Conference on Database and Application Security*. Ont., Canada, July 2001.

[9] J. Park, R. Sandhu, and G. Ahn. "Role-based Access Control on the Web," *ACM Transactions on Information and System Security*, 4(1), February 2001.

[10] E. Coyne. "Role Engineering," In *Proceedings of 1st ACM Workshop on Role-*

Based Access Control, Gaithersburg, MD, November 1995.

[11] ITU-T Recommendation X.509. Information Technology: Open Systems Interconnection – The Directory: Public-Key and Attribute Certificate Frameworks, 2000. ISO/IEC 9594-8:2001.

[12] S. Farrell and R. Housley. An Internet Attribute Certificate Profile for Authorization, PKIX Working Group, June 2001.

[13] Doug Allen, "Emerging Technology: DSML and DEN: Signs of Things to Come," *www.networkmagazine.com*, June, 2000.

[14] Rawn Shah, "DSML is the glue for future directories," *www.sunworld.com*, March, 2000.

[15] Organization for the Advancement of Structured Information Standards (OASIS). DSML V2 Specification. *https://www.oasis-open.org/committees/dsml/*, December 2001.

[16] Organization for the Advancement of Structured Information Standards (OASIS). SAML V 1.0 Specification. *http://www.oasis-open.org/committees/security/*, December 2001.