Role-Based Access Control for the Web

John F. Barkley, D. Richard Kuhn, Lynne S. Rosenthal, Mark W. Skall, and Anthony V. Cincotta,

National Institute of Standards and Technology Gaithersburg, Maryland 20899

ABSTRACT

Establishing and maintaining a presence on the World Wide Web (Web), once a sideline for U.S. industry, has become a key strategic aspect of marketing and sales. Many companies have demonstrated that a well designed Web site can have a positive effect on their profitability. Enabling customers to answer their own questions by clicking their way through Web pages, instead of dealing with operators and voice response systems, increases the efficiency of the customer interface.

One of the most challenging problems in managing large networked systems is the complexity of security administration. This is particularly true for organizations that are attempting to manage security in distributed multimedia environments such as those using World Wide Web services. Today, security administration is costly and prone to error because administrators usually specify access control lists for each user on the system individually.

Role-based access control (RBAC) is a technology that is attracting increasing attention, particularly for commercial applications, because of its potential for reducing the complexity and cost of security administration in large networked applications. The concept and design of RBAC is perfectly suited for use on both intranets and internets. It provides a secure and effective way to manage access to an organization's Web information. This paper describes a research effort to develop RBAC on the Web. The security and software components that provide RBAC for networked servers using Web protocols have been implemented and are described in this paper. The RBAC components can be linked with commercially available web servers, and require no modification of the server software.

Introduction

Establishing and maintaining a presence on the World Wide Web (Web), once a sideline for U.S. industry, has become a key strategic aspect of marketing and sales. Many companies have demonstrated that a well-designed Web site can have a positive effect on their profitability. Enabling customers to answer their own questions by clicking their way through Web pages, instead of dealing with operators and voice response systems, increases the efficiency of the customer interface. Companies are seizing the Web as a swift way to streamline - even transform their organizations.

More recently companies have begun using web technology to service the public as well as private and internal clients. Web sites are set up to segregate some information from the general public, providing it to only selected or "private" clients. Typically, public internet is cordoned off from the general public by having

user accounts and passwords. Additionally, Web sites are now running inside the company often created for and by employees. These internal private nets or "intranets" use the infrastructure and standards of the Internet and the World Wide Web but are cordoned off from the public Internet through firewalls.

The Web can be used as an inexpensive yet powerful alternative to other forms of communications. A plethora of corporate information (e.g., procedures, training materials, directories, forms) can be converted to electronic form and made available via the Web. With a single source for these materials the cost of maintenance is significantly reduced, while greatly simplifying the task of ensuring currency. Thus an objective of enterprise computing, creation of a company wide system irrespective of the underlying information technology infrastructure can be fulfilled.

Although the internet and intranets can offer great benefits to a company or government agency, security threats remain. To date net enthusiasts tend to focus on how to link people and businesses, not on using the network as a way to run and manage businesses securely. Although existing Web servers can effectively provide all or nothing access to a particular Web site and a number of popular Web servers can even provide fairly fine grained access control, they provide very primitive tools to administer these controls from the perspective of a single enterprise.

This paper describes the benefits of RBAC and an implementation of RBAC on the Web (RBAC/Web), and in particular as RBAC applies to an intranet computing environment. This will provide Web administrators with a capability for the first time to centrally administer and regulate user access to information in a manner that is consistent with the current set of laws, regulations, and practices that face their business today. Although this paper focuses on intranets, the benefits, concepts and implementation of RBAC/Web are also applicable to a company’s internet environment where restrictive access to information is desired.

RBAC Description

Role-based access control (RBAC) [1], [2], [3], [4], [5] is an alternative to traditional discretionary (DAC) and mandatory access control (MAC) policies that is attracting increasing attention [6], particularly for commercial applications. The principal motivation behind RBAC is the desire to specify and enforce enterprise-specific security policies in a way that maps naturally to an organization's structure. Traditionally, managing security has required mapping an organization's security policy to a relatively low-level set of controls, typically access control lists.

With RBAC, security is managed at a level that corresponds closely to the organization's structure. Each user is assigned one or more roles, where roles are based on the user's job responsibilities and competencies in the organization. Each role is assigned one or more privileges (e.g., information access, deletion, creation), see Figure 1. It is a user's membership into roles that determine the privileges the user is permitted to perform. Security administration with RBAC consists of determining the operations that must be executed by persons in particular jobs, and assigning employees to the proper roles.

Figure1: RBAC Relations

The RBAC framework provides for mutually exclusive roles as well as roles having overlapping responsibilities and privileges. For example, some general operations may be allowed by all employees, while other operations may be specific to a role. Role hierarchies are a natural way of organizing roles within an organization and defining the relationship and attributes of the roles. Complexities introduced by mutually exclusive roles or role hierarchies as well as regulating who can perform what actions, when, from where, in what order, and in some cases under what relational circumstances, is all handled by the RBAC software.

Separation of Duty

RBAC mechanisms can be used by a system administrator in enforcing a policy of separation of duties. Separation of duties is considered valuable in deterring fraud since fraud can occur if an opportunity exists for collaboration between various job related capabilities. Separation of duty requires that for particular sets of transactions, no single individual be allowed to execute all transactions within the set. The most commonly used examples are the separate transactions needed to initiate a payment and to authorize a payment. No single individual should be capable of executing both transactions. The system administrator can control access at a level of abstraction that is natural to the way that enterprises typically conduct business. This is achieved by statically and dynamically regulating users' actions through the establishment and definition of roles, role hierarchies, relationships, and constraints.

We define static separation of duty to mean that roles which have been specified as mutually exclusive cannot both be included in a user's set of authorized roles. With dynamic separation of duty, users may be authorized for two roles that are mutually exclusive, but cannot have both roles active at the same time. In other words, static separation of duty enforces the mutual exclusion rule at the time an administrator sets up role authorizations, while dynamic separation of duty enforces the rule at the time a user selects roles for a session.

The mutually exclusive roles for a given role and the **Static Separation of Duty** property can be specified as follows:

mutually-exclusive-authorization(r:roles) = {the list of roles that are mutually exclusive with role

"r"}

A user is authorized as a member of a role only if that role is not mutually exclusive with any of the other roles for which the user already possesses membership:

Dynamic Separation of Duty places constraints on the simultaneous activation of roles. The mutually exclusive roles for a proposed active role is specified as follows:

mutually-exclusive-activation(r:roles) = {the list of active roles that are mutually exclusive with the proposed role "r"}

A subject can become active in a new role only if the proposed role is not mutually exclusive with any of the roles in which the subject is currently active:

Role Administration and Visualization

The roles are established, manipulated and viewed using the RBAC/Web Admin tool. The Admin tool allows system administrators to create and define roles, role hierarchies, relationships and constraints. Once the RBAC framework is established for the organization, the principal administrative actions are the granting and revoking of users into and out of roles as job assignments dictate. These maintenance tasks are easily performed using the Admin tool.

Additionally, the Admin tool is being enhanced to utilize the Virtual Reality Modeling Language (VRML, pronounced 'vermal'). VRML is an interactive, inter-networked, 3D graphics language for the Web. It is used to represent graphics, test, sound, and links to other content as either a static or dynamic picture on the Web. The inclusion of VRML into RBAC lets system administrators use an interactive computer model to check and validate the role structure, relationship, and privileges. Being able to view and interact with complex models, allows the administrator to identify conflicts, eradicate flaws and improve the implementation early in the RBAC setup.

The VRML component will enable authorized users to navigate the RBAC database, finding and linking roles, and displaying attributes and graphics associated with those roles. By presenting a 3D model of established roles, the user can easily see which roles are mutually exclusive as well as the hierarchical

structure of related roles and conflicts between roles (see Figure 2). VRML's navigational controls allows the user to interactively 'walk-through' and manipulate the view perspective of the 3D model, known as a scene graph. For example, the scene graph can be rotated to show the 'backside' of the graph where role relationships may have been obscured when viewed as a 'flat', 2D graph. To improve readability, clarity and flexibility, the role hierarchy is organized into layers, where each layer contains another level of detail. By 'clicking' on a role, the role opens to reveal the next layer of related roles or information about the role, e.g., the privileges associated with that role or a user membership list.

Figure 2: VRML Bank example

RBAC Example

Consider the branch office of a bank. In this environment, there are roles such as branch manager, teller, and account representative, as illustrated in Figure 2.

The graph structure shows role hierarchy. The role *financial_advisor* inherits the role *account_rep*. An individual authorized for the role *financial_advisor* is permitted to perform all of the operations permitted to an individual authorized for the role *account_rep*. Thus, an individual in the role of *financial_advisor* is able to create and remove accounts. Because account representatives, branch managers, internal auditors, and tellers are all employees of the bank, their corresponding roles inherit the employee role.

In Figure 2, the role *account_rep* is highlighted, appearing as a dark sphere, in order to show the other role relationships for *account_rep*. The roles teller and *account_holder* are shown as yellow rectangular solids to

indicate that these roles have a "Dynamic Separation of Duties" (DSD) relationship with the role *account_rep*. This relationship is a conflict in interest relationship indicating that an individual acting in the role of *account_rep* cannot also be acting in either of the roles of *account_holder* or *teller*. The policy of the bank is that an account representative, an employee of the bank, can have an account in the bank but such an individual may not simultaneously process their personal account while processing accounts of others. Likewise, because a teller has an open cash drawer that must balance when closed, an individual acting in the role of *account_rep* and sitting at a desk away from a teller's window is not permitted to simultaneously act in the role of *teller* even if authorized for that role.

The role internal_auditor is shown in a red hexahedron to indicate that this role has a "Static Separation of Duties" (SSD) relationship with the role *account_rep*. The SSD relationship is also a conflict of interest relationship like the DSD relationship but much stronger. If two roles have a DSD relationship, then they may both be authorized for an individual but that individual may not act in both roles simultaneously. If two roles have a SSD relationship, then they may not even be authorized for the same individual. In this example, the policy of the bank is that there is a fundamental conflict of interest between the roles of internal_auditor and *account_rep*. Thus, these two roles may never be authorized for the same individual.

The new version of the Admin tool using VRML will allow us to represent conflicts of interest and other relationships in a more natural way and view the scene from an infinite number of viewpoints. VRML allows complex 3D objects to be created for this purpose. The user can 'enter' a selected role and explore several levels of detail (i.e., information) associated with that role. In addition, the sound capabilities of VRML can be utilized to give audio warnings when roles are used which cause conflicts of interest or other problems, or when improper procedures are used.

RBAC for World Wide Web Applications

Role Based Access Control (RBAC) for the World Wide Web (RBAC/Web) is an implementation of RBAC for use by World Wide Web (Web) servers. Because RBAC/Web places no requirements on a browser, any browser that can be used with a particular Web server can be used with that server enhanced with RBAC/Web. RBAC/Web is implemented for both UNIX (e.g., for Netscape, NCSA, CERN, or Apache servers) and Windows NT (e.g., for Internet Information Server, WebSite, or Purveyor) environments.

Components of RBAC/Web are shown in Table 1. RBAC/Web for UNIX uses all of the components in Table 1. Because built-in NT security mechanisms are closely compatible with RBAC, the NT version uses only the Database, Session Manager, and Admin Tool components. RBAC/Web for NT requires no modification of Web server internals or access to source code. With RBAC/Web for UNIX, there are two ways to use RBAC/Web with a UNIX Web server.

The simplest way is by means of the RBAC/Web CGI. The RBAC/Web CGI can be used with any existing UNIX server without modifying its source code. RBAC URLs are passed through the Web server and processed by the RBAC/Web CGI. RBAC/Web configuration files map URLs to file names, while providing access control based on the user's roles. Installation of the RBAC/Web CGI is similar to the installation of the Web server.

While the RBAC/Web CGI is relatively simple to install and use, it is not as efficient as performing access control directly in the Web server. The other way to use RBAC/Web is to modify the UNIX Web server to call the RBAC/Web API to determine RBAC access. A URL is configured as an RBAC controlled URL by means of the Web Server configuration files that map URLs to file names.

Some Web servers for a UNIX environment, such as Netscape and Apache, divide their operation into steps and provide the capability for each step to be enhanced or replaced by means of a configuration parameters. This allows Web server operation to be modified without having to change the server's source code. For these Web servers, the RBAC/Web API can be integrated by simply providing the appropriate calling sequence and modifying configuration parameters.

Database	Files that specify the relationship between users and roles, the role hierarchy, the constraints on user/role relationships, current active roles, and relationship between roles and operations.
Database Server	Hosts the authoritative copies of the files which define relationships between users and roles, the role hierarchy, and the constraints on user/role relationships. These files are created and maintained by the Admin Tool. When changes are made these files, the Database Server notifies the Web Servers to update their cached copies.
API Library	A specification which may be used by Web servers and CGIs to access the RBAC/Web Database. The API is the means by which RBAC may be added to any Web server implementation. The API Library is a C and Perl library which implements the RBAC/Web API.
CGI	Implements RBAC as a CGI for use with any currently existing Web server without having to modify the server. The RBAC/Web CGI uses the RBAC/Web API.
Session Manager	Manages the RBAC Session. The RBAC/Web Session Manager creates and removes a user's current active role set (ARS).
Admin Tool	Allows server administrators to create users, roles, and permitted operations; associate users with roles and roles with permitted operations; specify constraints on user/role relationships; and maintain the RBAC Database. Administrators access the RBAC/Web Admin tool by a Web browser

Table 1: RBAC/Web Components

Authentication

RBAC is an access control mechanism that can be used in conjunction with existing Web authentication and confidentiality services. These include username/password, Secure Socket Layer (SSL), Secure HTTP (SHTTP), and Private Communication Technology Protocol (PCT). User identification information is passed to RBAC/Web by the Web server. It is the responsibility of the Web server to authenticate user identification information and provide confidential data transmission as configured by the Web server administrator.

End-User Use Scenario

End-user interaction with a Web server enhanced with RBAC/Web is basically the same when requesting URLs whose access is not controlled by RBAC/Web (see Figure 3). However, before access to a URL controlled by RBAC is permitted, end-users must establish an RBAC session. In establishing the RBAC session, end-users choose and/or are assigned a current active role set (ARS). The ARS determines the

permitted operations that the end-user can perform on RBAC controlled URLs. The ARS remains in effect until the end-user establishes a new ARS. It is the ARS which constitutes the RBAC session.

A user may be assigned roles which have DSD relationships. If this is the case, the Session Manager enables users to choose the subset of their assigned role set that they would like to use in the session. Users are presented with a list of subsets which do not violate any DSD relationships and ask to choose. In order to minimize the number of choices, the subsets in the list, taken from the set of all possible subsets of a user's assigned roles, contain the largest subsets which do not violate any DSD relationships. Once the choice is made, the RBAC session is established with all authorized roles (i.e., assigned roles along with all roles which the assigned roles inherit) being placed in the ARS. If there are no DSD relationships among the roles assigned to a user, then the RBAC session is automatically established with all authorized roles in the ARS.

Figure 3: RBAC/Web Use

Generally, an RBAC session requires an authenticated end-user. If authentication is removed from an end-user, access to RBAC controlled URLs is denied. However, end- user authentication and the establishment of an RBAC session are completely separate operations. This is so that RBAC/Web can work with any authentication mechanism.

Conclusions

For the Web to reach its full potential as a means for enterprise computing, access control mechanisms must be in place that can regulate user access to information in a manner that is consistent with the current set of laws, regulations, and practices that face businesses today. The purpose of RBAC/Web is to provide this access control service. This makes it possible to use the Web for new and more sophisticated applications--to allow access to information and other resources that would otherwise not be possible given the existing and emerging threat environment.

One of RBAC's greatest virtues is the administrative capabilities it supports. The administration of authorization data is widely acknowledged as an onerous process with a large and reoccurring expense. Under RBAC, users are granted membership into roles based on their competencies and responsibilities. User membership into roles can be revoked easily and new memberships established as job assignments dictate. With RBAC, users are not granted permission to perform operations on an individual basis, rather, operations are associated with roles. Role association with new operations can be established as well as old operations deleted as organizational functions change and evolve. This basic concept has the advantage of simplifying

the understanding and management of privileges: roles can be updated without having to directly update the privileges for every user on an individual basis.

RBAC/Web provides the advantages of role-based access control for intranet and internet environments, and can be incorporated into existing systems with no modification to server code, making it portable to virtually all Web servers. For more information on RBAC and RBAC/Web, see [http://csrc.nist.gov/groups/SNS/rbac/].

References

[1] D. Ferraiolo and D. R. Kuhn. Role-based access control. In 15th National Computer Security Conference. NIST/NSA, 1992.

[2] D. F. Sterne. A tcb subset for integrity and role-based access control. Appendix - RBAC Formal Description . In 15th National Computer Security Conference. NIST/NSA, 1992.

[3] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. IEEE Computer, 29(2), February 1996.

[4] S. H. von Solms and I. Van der Merve. The Management of computer security profiles using a role oriented approach. Computers and Security, 13(8), 1994.

[5] 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.

[6] R. Sandhu, E. J. Coyne, and C. E. Youman, editors. Proceedings of the First ACM Workshop on Role Based Access Control. ACM, 1996.

Disclaimer

Because of the nature of this report, it is necessary to mention vendors and commercial products. The presence or absence of a particular trade name product does not imply criticism or endorsement by the National Institute of Standards and Technology, nor does it imply that the products identified are necessarily the best available.