(5) RBAC Transition

Charles E. Youman

SETA Corporation
cyouman@seta.com

1.0 Introduction

The topic of role-based access control (RBAC) transition considers how to migrate existing systems from their current state (with no RBAC capabilities) to a state that includes an RBAC mechanism. The workshop participants did not think that RBAC would exist in a vacuum. The participants did not expect RBAC to replace existing access control mechanisms such as discretionary access controls (DAC) and mandatory access controls (MAC). To be accepted, RBAC needs to co-exist with the current access control mechanisms. RBAC is also not expected to be the only applications level access control mechanism. For example, RBAC should be capable of co-existing with a Chinese wall access level applications control policy.

The views of the workshop are consistent with how some Government and commercial security standards are evolving. The U.S. Department of Defense Goal Security Architecture states "although most current information systems support only one information security policy at a time, there has long been a desire by users to operate simultaneously ... under multiple security policies" [DOD94, p. 2-3]. As an example of a commercial security standard, the Common Object Request Broker Architecture (CORBA) Security Specification includes a requirement for flexibility of security policy that includes a choice of access control policy [OMG95a, p. 19]. However, this view is not universal. For example, the CS3 protection profile of the Common Criteria "is applicable to networked or distributed environments only if the entire network operates under the same constraints and resides within a single management domain" [COMM96, p. 24].

To illustrate some of the transition issues that must be addressed in order to successfully incorporate RBAC into existing systems, the discussion will follow a sequence of events where an RBAC mechanism is installed on an existing system, an application that uses RBAC is installed, and a second RBAC application is installed. The sequence of steps organizes the discussion of issues in RBAC transition and is not necessarily the set of steps that would be taken in an actual implementation of RBAC.

ACM RBAC Workshop, MD, USA
© 1996 ACM 0-89791-759-6/95/0011 $3.50

2.0 Existing System

Implementation of RBAC requires that certain features be present in the existing system. First, there needs to be an identification and authentication mechanism. Second, a single userid needs to be assigned to each user of the system. Implementing separation of duties (a common RBAC constraint) requires that each user of the system possesses one and only one userid. Third, other features, such as access control for objects visible to the operating system, object reuse, and an audit mechanism, common to operating systems that meet level C2 of the Orange Book [DOD85], are also assumed to be present. Some implementations of RBAC may require additional operating system features. A paper [SMIT94] describes a banking application developed for the Republic of China using RBAC. It was built on top of a B1 operating system. The MACs were used to create a category that contains the metadata about the roles defined in the system and access to the role metadata is limited to system administrators.

It is also assumed that the existing security mechanisms are configured so that the system satisfies the security policy that has been established for the system. Installing RBAC will not repair any security flaws previously present in the system.

3.0 RBAC Mechanism Added

One requirement for the transition process is that existing applications will run without modification. The justification for this requirement is that if there is to be widespread acceptance of RBAC, it must be possible to implement a phased transition. Existing applications should be able to run without modification along with applications that use RBAC. Over an extended period of time, existing applications can be modified to utilize RBAC.

A second requirement for the transition process is that installation of the RBAC mechanism should not introduce security weaknesses into the system. In other words, if the system was previously in a secure state, then installing an RBAC mechanism should leave the system in a secure state.

A related RBAC transition requirement arises when an application that uses RBAC is run but no RBAC mechanism is installed, then the access requested by the RBAC application should be denied. It may not be apparent to the operating system that an application utilizes RBAC until the application sends a request for service to the RBAC mechanism.

Ideally, it would be desirable to add RBAC capabilities to existing applications without modification. This would also make it possible to integrate commercial off-the-shelf (COTS) products into a system-wide RBAC mechanism. This is similar to the CORBA security architecture view of application portability, which states that "an application object should not need to be aware of security, so it can be ported to environments that enforce different security policies and use different security mechanisms" [OMG95a, p. 20].

In a large distributed system, one question that needs to be addressed is where to locate the outer boundary or horizon that establishes whether a role is visible to a user. Since users are mapped to roles, one constraint on the location of the outer boundary would be that it not extend past the point where there is some minimum level of confidence in the identity of the user. While generally the identity of users needs to be established in order to determine what roles they are authorized, in some circumstances it may be desirable to permit anonymous users or users whose identity may be suspect to assume a role. These would typically be roles with modest capabilities such as access to publicly available information. It is clear, however, that access by users whose identity cannot be authenticated would not extend to users assuming the role of security administrator. The integrity of role definitions and role assignments needs to be maintained.

4.0 Install RBAC Application

The steps that need to be accomplished to install an RBAC application include identifying privileges needed to run the application, defining roles, applying constraints, and authorizing users to assume their roles.

On a single system, RBAC applications may co-exist with applications that don't support RBAC. On such a system, privileges may need to be granted directly to users (as well as to roles) in order to support the non-RBAC applications. Under these circumstances, it may be possible for a privilege to be granted to both an RBAC role and directly to a general user. If the user is allowed to assume a role that does not include this privilege, it may allow the user to circumvent RBAC constraints intended to create a separation of duties. However, a similar problem can exist entirely within RBAC if the roles are not properly defined to provide the desired separation of duties or if an individual can have multiple userids.

5.0 Install a Second RBAC Application

One issue that arises once there is more than one application using RBAC is whether role names apply only to one application, or whether the names have to be unique throughout the domain of the RBAC mechanism. For example, several applications may have a role called *clerk* where the privileges associated with the role are different.

A related issue is whether there is a single role database for the organization or a role database for each application. Having a common database of role definitions may simplify role administration and allow multiple applications to be defined as part of a single role. However, it is also possible that the horizon for the applications may be different, which would complicate having a single role database. It also may be desirable to allow each installation the latitude to determine how many role databases exist.

References

[COMM96] Common Criteria Editorial Board, *Common Criteria for Information Technology Security Evaluation*, Version 1.0, January 1996. Available from: http://csrc.nist.gov/nistpubs/cc/read_me.cc1.

[DOD85] U.S. Department of Defense, *Trusted Computer Systems Evaluation Criteria*, DOD 5200.28-STD, Washington, DC, December 1985.

[DOD94] U.S. Department of Defense, Defense Information Systems Agency, Center for Architecture, *Department of Defense Technical Architecture Framework for Information Management, Volume 6: Department of Defense (DoD) Goal Security Architecture*, Version 2.0, Washington, DC, 30 June 1994.

[OMG95a] AT&T Global Information Solutions Co., Digital Equipment Corporation, Expersoft Corporation, Groupe Bull, Hewlett-Packard Company, International Business Machines Corporation (in collaboration with Taligent Inc.), International Computers Limited, Novell, Inc., Siemens Nixdorf Informationssysteme AG, Sunsoft, Inc., Tandem Computer (in collaboration with Odyssey Research Associates, Inc.), and Tivoli Systems, Inc., *CORBA Security*, OMG Document Number 95-12-01, Framingham, MA: Object Management Group, December 1995. Available from: ftp://ftp.omg.org/pub/docs/1995/95-12-01.ps.

[SMIT94] Barbara Smith-Thomas and Wang Chao-Yeuh, "Implementing Role Based, Clark-Wilson Enforcement Rules in a B1 On-Line Transaction Processing System," *Proceedings of 17th National Computer Security Conference*, Baltimore, MD: National Institute of Standards and Technology, 11-14 October 1994, pp. 56-65.