DRAFT

Role Based Access Control Implementation Standard

Version 0.1

January 2006

DRAFT

Contents

1

Introduction

This standard describes implementation requirements for RBAC systems. The functional specifications of the RBAC components defined in the RBAC standard are described in this standard to promote their implementation in a standard manner. It is intended for (1) software engineers and product development managers who design products incorporating access control features; and (2) managers and procurement officials who seek to acquire computer security products with features that provide access control capabilities in accordance with commonly known and understood terminology and functional specifications. Adherence to this standard will provide a basis for the interchange of data and functional interoperability among services and applications.

1 Scope

The RBAC standard's section on System and Administrative Functional Specification specifies the features that are required of an RBAC system. These features fall into three categories, administrative operations, administrative reviews, and system level functionality. This RBAC implementation standard specifies how these features are to be implemented.

2 Conformance

This standard specifies the packaging of features through the selection of functional components and feature options within a component, beginning with a core set of RBAC features that must be included in all packages. Other components that may be selected in arriving at a relevant package of features pertain to role hierarchies, static constraints (Static Separation of Duty), and dynamic constraints (Dynamic Separation of Duty).

In addition to making reference to packages of the features described in the RBAC standard, this standard includes use cases that must be supported by compliant designs.

2.1 Components

The RBAC standard's system and administrative functional specification contains descriptions of functions for four RBAC components (see section 6.1). A compliant design will contain a set of components selected from the following:

1. Core RBAC,
2. Hierarchical RBAC,
3. Static Separation of Duty (SSD) Relations, and
4. Dynamic Separation of Duties (DSD) Relations.

Several options exist for a product design to be compliant with this implementation standard. All options include Core RBAC. The options are defined as combinations of Core RBAC with one or more of the remaining three of the RBAC components, as illustrated in Table 1.

Table 1. Options for Inclusion of Components

	Option							
Component	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Core RBAC	●	●	●	●	●	●	●	●
Hierarchical RBAC		●				●	●	●
Static Separation of Duty (SSD) Relations			●		●	●		●
Dynamic Separation of Duties (DSD) Relations				●	●		●	●

2.2 Use Cases

2.2.1 Operational

2.2.2 Administrative

2.3 Completeness

Completeness with respect to meeting this standard refers to the set of RBAC components identified in section 2.1. Within each option, or combination of components, the commands and functions listed in section 6.1 must be included.

2.4 Correctness

2.5 Interoperability

Interoperability is to be achieved through common data specifications and common interfaces.

3 Normative References

American National Standard *ANSI INCITS 359-2004* Role Based Access Control

4 Terms and Definitions

The following terms have specialized meanings within this standard.

Component: A *Component* refers to one of the major blocks of RBAC features, core RBAC, hierarchical RBAC, SSD relations, and DSD relations.

Feature: A *Feature* is loosely defined as an item contained within an RBAC design to provide functionality.
Object: As used in this standard, an *object* can be any system resource subject to access control, such as a file, printer, terminal, database record, etc.
Operation: An *operation* is an executable image of a program, which upon invocation executes some function for the user.
Permission: A *Permission* is an approval to perform an operation on one or more RBAC protected objects.
Role: A *role* is a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role.
User: A *user* is defined as a human being. Although the concept of a user can be extended to include machines, networks, or intelligent autonomous agents, the definition is limited to a person in this document for simplicity reasons.

5 Symbols and Abbreviated Terms

6 Requirements

The requirements for a compliant design are focused on support for a set of use cases. For each set of components (see section 6.1) included in a design, the corresponding use cases (see section 2.2) must be supported.

6.1 Components

The RBAC components and their corresponding functions are reproduced here (section numbers from the RBAC standard have been preserved, as "STD-6.x"). For a design to be compliant, for each component addressed, the below-listed commands and functions (e.g., STD-6.1.1, STD-6.1.2) must be provided.

STD-6.1 Core RBAC
- STD-6.1.1 Administrative Commands for Core RBAC
- STD-6.1.2 Supporting System Functions for Core RBAC
- STD-6.1.3 Review Functions for Core RBAC
- STD-6.1.4 Advanced Review Functions for Core RBAC

STD-6.2 Hierarchical RBAC
- STD-6.2.1 General Role Hierarchies
  - STD-6.2.1.1 Administrative Commands for General Role Hierarchies
  - STD-6.2.1.2 Supporting System Functions for General Role Hierarchies
  - STD-6.2.1.3 Review Functions for General Role Hierarchies
  - STD-6.2.1.4 Advanced Review Functions for General Role Hierarchies
- STD-6.2.2 Limited Role Hierarchies

6.2 Data Model

6.3 Interfaces

6.3.1 Policy Decision Point

The concept of Policy Decision Point (also known as Access Control Decision Function) is a locus where policy rules have been resolved, evaluated, and combined to yield a binary value for interpretation by a Policy Enforcement Point. The OASIS XACML standard defines Policy Decision Point and its implementation using the XACML language.

6.3.2 Policy Enforcement Point

The concept of Policy Enforcement Point (also known as Access Control Enforcement Function) is where a policy decision is used to grant or deny access to a protected resource. A Policy Enforcement Point typically exists within an application.

7 Normative Annexes

8 Informative Annexes

8.1 Discussion on Two Types of Roles: Basic and Functional

Provided by US Department of Veterans Affairs, Veterans Heath Administration

8.1.1 Purpose

"Basic"[1] roles being defined in ASTM and elsewhere provide a means to enforce "connect" authorizations for authenticated users independent of determining functional roles and authorizing detailed operations on protected information objects. Basic roles "firewall" information resources by effectively managing which applications and workflows are permitted to a user in the first place. Basic roles support service-based architectures where it is desirable to centrally manage user access to protected resources.

[1] Basic roles are defined in Bernd Blobel's *Analysis, Design and Implementation of Secure and Interoperable Distributed Health Information Systems* (2002). These are alternatively called static roles, or role groups by other sources.

It is advantageous in these environments to place the concepts and definitions of "functional" roles into a context that includes basic roles. The basic role can be considered to be a type of prerequisite role, i.e., supporting a user authorization that occurs before other roles can be activated.[2]

8.1.2 Discussion

In Figures 1 and 2 in the RBAC standard, user activation of roles follows once a session is established. The session activates a subset of the user's assigned roles (session roles). This subset of roles consists of functional (dynamic) roles. They are dynamic because they are activated in the context of the session and user session attributes. They contain the permissions that a user has available once the session is established and the roles are activated. Implementations of such roles are typically managed in applications, directories, and attribute certificates.

In establishing the session, however, there is an implicit assumption that the users in Figures 1 and 2 are in fact authenticated users of the system who are authorized to invoke certain permissions, such as opening a session. Thus, these authenticated users have implicit or explicit permissions to initiate sessions. Functional role activation (session roles) cannot occur until the session is established, and authorization to establish the session may occur outside of the application authorization functions. To accomplish this basic connect function, the user would possess, in addition to authentication information, some set of basic (static) roles that would be prerequisites to a user's being authorized to "connect" to the task or workflow containing the session (functional) roles. An access control enforcement function would have the responsibility to grant or deny the session based on the basic role.

Basic roles therefore contain permission to participate in specific workflows or tasks that require access to an information object such as a database that is managed by session oriented functional roles. Basic roles allow basic "connect" permission to task-related information stores. Basic roles would be typically managed in identity certificates or directories.

8.1.3 Basic Roles vs Organizational Roles

Organizational roles are those roles that reflect the organization chart of an enterprise.[3] In some cases, organizational roles and basic roles may be conterminous. The distinction between the two is that organizational roles are taken from the organizational structure. Basic roles may or may not also be organizational roles.

8.1.4 Roles in ASTM Healthcare Policy and Standard Guide

Basic Role groups may be found as categories of subscribers for healthcare certificates in the ASTM digital certificate policy [E 2212]). This policy provides for roles and

[2] A basic role could also serve as a functional role, should the security policy permit.

[3] See David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli, *Role-Based Access Control* (2003).

credentials for healthcare organizations to use that are part of non-critical extensions to an X.509 v3 PKI certificate. ASTM views these roles as basic roles. They are "static" since they are part of the identity certificate, and exist as long-term attributes of the user.

The ASTM Standard Guide for Information Access Privileges to Health Information [E 1986] represents healthcare basic roles suitable for use in this standard. Some healthcare basic role examples include: Physician, Pharmacist, Advanced Practice Registered Nurse, and Ward Clerk. These are basic roles suitable for session connect privilege that do not necessarily specify what the user can do once connected.

While these ASTM standards are oriented to healthcare, the concepts pertain to any business area.

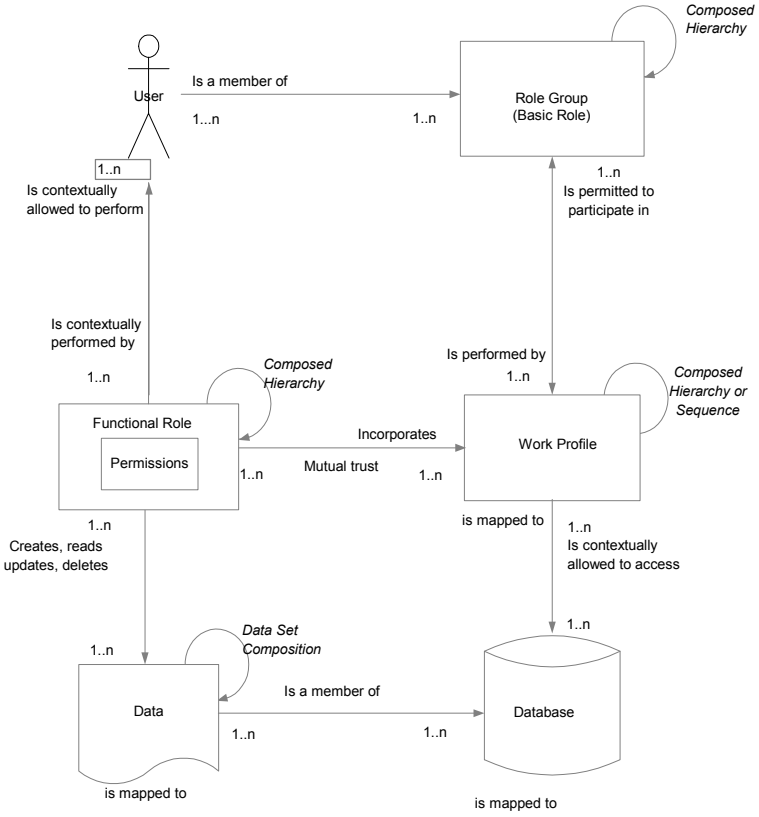

Figure 1: Role Engineering Model

Figure 1[4] illustrates the relationships between role groups (basic roles, static roles) work profiles, and functional roles (groups of permissions) consistent with the ASTM healthcare policy and the RBAC standard.

Role groups (basic roles) place people within an organization's personnel (not necessarily organizational) structure into categories of personnel warranting differing levels of access control. Role groups allow users to participate in the organization's workflow (e.g., tasks) by job, title, or position but do not specify detailed permissions on

[4] Adapted from Health Level 7 Security Technical Committee

specific information objects. As stated earlier, role groups can allow a user to "connect" to a resource but do not necessarily grant finer-grain authorizations on protected information objects.

As depicted in Figure 2 (extracted from Figure 1), role groups define what specific work profiles users are allowed to perform, while functional roles define what authorizations are needed by an entity to access protected information technology or application resources.

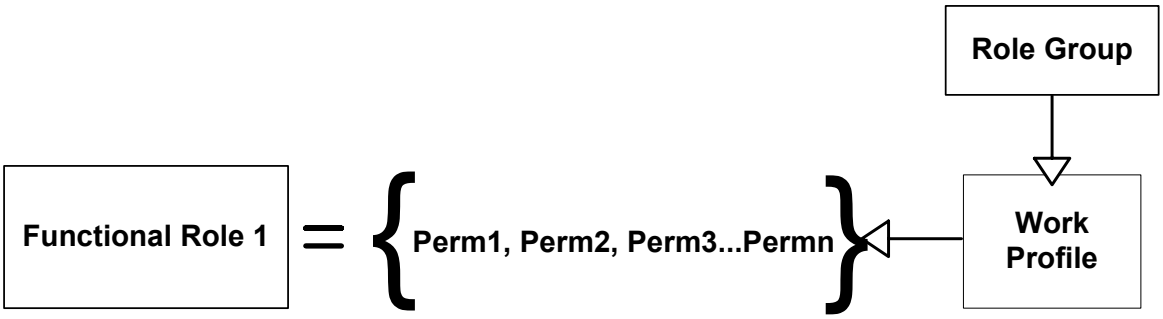

Figure 2: Role Groups and Functional Roles

8.1.5 Conclusion

Basic roles can provide the basic connect permissions that precede the user activation of roles described in the core RBAC and hierarchical RBAC models.

(Best practices should be provided in an informative annex, e.g., consistency checking)