Migrating to Role-Based Access Control

Kami Brooks
AverStar Inc.
6305 Ivy Lane, Suite 701, Greenbelt, MD 20770
kbrooks@averstar.com

Abstract

This project seeks to provide a single, GUI based security management interface for an existing, highly complex information systems environment. Among the identified requirements and goals for this project are, the use of Commercial Off-the-Shelf software, and the implementation of a foundation for an RBAC based approach to security management. This paper presents an overview of the RBAC salient issues that have been surfaced by initial efforts. This paper also highlights some of challenges faced in migration from an existing environment that has been developed over time and is largely segmented in both user communities and support groups to a centralized RBAC environment.

Keywords: Role-Based Access Control, migration, enterprise systems management, Tivoli Management Environment, security management

1. Introduction

The will support multiple computing centers that provide data processing, storage, and systems management on a fee basis. The various technologies encompassed by the collective sites are reflected in The Project Road Map, Figure 1. The sponsor is charged with the design and enforcement of security policy for the sixteen computing centers. This project represents a multi-year effort to achieve a single, graphical user interface for security management of the various operating systems, application servers, and the associated communications infrastructure. The project requires the use of Commercial Off-The-Shelf (COTS) products. Numbered among the goals is the establishment of a Roles-Based Access Control (RBAC) based security management approach capable of enforcing enterprise level security policies while permitting localized enhancements. Successfully achieving an RBAC based approach is expected to enhance:

- Information assurance through consistent security management.
- Auditing and situation response through improved understanding of how information resources relate to enterprise functions.
- The ability to define and support the principle of least privileged through increased granularity and flexibility in access management.
- Secure delegation of selected administrative functions.

The rest of this paper is organized as follows: Section 2 provides an overview of the managed environment and the products used in this effort. Section 3 discusses efforts to achieve user identification, both uniquely identified individuals and processes. Section 4 discusses selected constraints that have been implemented as user attributes instead of roles. Section 5 discusses techniques used to achieve the concepts presented in Administrative RBAC '97 (ARBAC97). Section 6 discusses the initial offering of a migration tool. Section 7 provides conclusions and references.

2. Environment Overview

The Managed Environment

The project is targeted to support more than fifteen computing centers that operate autonomously. Despite independent operation, all are bound by the same enterprise level security requirements. These activities represent wholly owned subsidiaries of a parent corporation. Corporate security policies establish minimum requirements and are typically augmented or customized by the local Security Manager to meet local or customer needs. The combined activities support in excess of 600,000 user accounts for a wide variety of systems, services, and communications networks.

The project will address three general levels of security management as depicted in Figure 1. The first phase of this effort, to be implemented in fall 1999, will:

- Introduce the project and its goals to the target environments.
- Add the TME-10 User Administration product to the existing environment.

Figure 1. Project Road Map

Level III, Communications	*Perimeter Services* TCP Wrappers *Filters and ACLs on Routers and Firewalls* *Modem pool dial-in and dial-out access*		*Authentication Services* PKI, Token Devices	
Level II, Infrastructure Services	*Applications* *Office automation such as: word processing, presentation, spreadsheet, and productivity applications*	*Email Services* Lotus Notes Servers, MS Exchange Servers	*Data Services* Oracle Servers MS SQL Servers Directory Servers	*Inter-Comm Services* IBM eNetwork Netscape SuiteSpot Microsoft ISS
Level I, Operating Systems	*Unix* HP-UX, Solaris, SVR4	Windows NT	AS/400	OS/390	Tandem

- Provide a subset of the capabilities described in this document, including:
  - User account management at the operating system level for Unix and Windows NT.
  - A centralized mechanism for user identification.
  - A profile design tool to assist in the identifying roles.

Future project phases will expand the number of managed operating systems and address management of additional services.

The Tivoli Management Environment

An existing installed based of the Tivoli Management Environment (TME) will provide the mechanism for achieving the project goals. The term "TME " describes a suite of products integrated with the Tivoli Management Framework (TMF) by Tivoli Systems, Inc. Key products for this project are the TME-10 Tivoli Management Framework, TME-10 User Administration, and TME-10 Security Management. A subscripted '*T*' is used to differentiate between references to TME mechanisms, such as a Security Management Role$_T$ and RBAC terminology.

Tivoli Management Framework

The TMF provides a graphical desktop, the ability to abstract system specific implementation issues, and a secure communications architecture. [TivF98] Significant to this effort are the following TMF constructs:

Tivoli Administrators

Each Tivoli Administrator$_T$ represents a description of a set of specific management activities authorized within the TME. The Administrator$_T$ authority is given to one or more individuals by associating the individual's login ID to the Administrator$_T$. Constraints on Administrators$_T$ include:

- Minimal membership cardinality of one. At least one login ID must be associated to an Administrator$_T$.
- Administrators$_T$ roles are mutually exclusive with respect to login ID. A single login ID may only be associated on one Administrator$_T$.

Policy Regions

The Policy Region is a core design element that represents a collection of managed resources. Pertinent constraints associated with Policy Regions include:

- Policy Regions may contain other Policy Regions providing the ability to create ARBAC hierarchies.
- Administrators$_T$ are assigned authority to conduct specific management activities at the Policy Region level by assigning Administrator Roles$_T$.
- Administrators$_T$ exercise the same Administrator Roles$_T$ over all managed resources contained within the region.
- Once assigned, Administrative Roles$_T$ flow down through all other subordinate Policy Regions.
- A managed device may only exist in a single Policy Region, but can be subscribed (the recipient of) to many management profiles.

Profile Managers

Tivoli Profiles store application specific information used to manage a set of managed devices. The TMF does not provide any specific profile types but does provide Profile Managers to contain the profiles provided by other integrated applications. Profile Manager constraints include:

- All profiles must be created within the context of a Profile Manager.
- Managed resources are subscribed to Profile Managers (as opposed to the Profile itself). Once subscribed, the managed resource is a target of all Profile Manager contained Profiles.
- A single managed resource may subscribe to multiple Profile Managers
- A Profile Manager can subscribe to another Profile Manager enabling of Role Hierarchies.

Managed Resources

In the most generic sense, a managed resource is a Tivoli resource contained inside a Policy Region and, thereby, subject to Tivoli policies. When discussed in this paper, a managed resource or device refers either to a physical object -- a Unix Server, a directory or a file-- or to a logical object -- an NIS+ Server, an Oracle Server-- that are the target of management actions. Managed resources represent objects to which permissions are applied.

Policies

Policies are user configurable programs applied to managed resources. These programs establish what is or is not permissible when a pre-determined management activity is attempted. Default and validation policies permit senior Administrators$_T$ to control the actions of junior Administrators$_T$.

- Default policies aid or control the creation of managed resources. They can supply pre-determined attribute values or ensure that minimal constraints are met.
- Validation policies ensure that resource properties meet specific constraints and prevent the creation or update of managed resources in a way that violates defined constraints.

User Administration

TME-10 User Administration is a profile-based application that provides user, group, and host management services at the operating system level. [TivA98] Significant to this discussion are:

UserProfiles

A UserProfile is a collection of user account records. A UserProfile can be distributed to multiple (subscribed) managed devices with one management action. Each record can contain information about General User Identity, Unix accounts, Windows NT accounts, NetWare accounts, and Security Management Group memberships.

GroupProfiles

A GroupProfile is a collection of Unix group records. While Unix groups are important to RBAC, they are not handled directly through the GroupProfile. Instead groups (both Unix and Windows NT) are handled within UserProfiles and Security Management Group records.

Security Management

TME-10 Security Management provides a solution for enterprise role-based distributed security management. [TivS98] Significant to this discussion are the following Security Management constructs:

Resource Records

Resource Records identify system resources such as programs, files, connection modes, services and more. The Resource$_T$ definition includes default access permissions such as time-of-day restrictions and access audit controls. Resources$_T$ are subsequently associated to Security Management Roles$_T$ that provide role specific permissions for the Resource$_T$

Roles

Roles$_T$ define a set of capabilities (Resources$_T$) required to carry out a given job. Roles$_T$ can be nested, with the child inheriting the capabilities of the parent. The child Role$_T$ can define new capabilities, such as access to new Resources$_T$; can increase inherited capabilities, such as adding write access to an inherited read only Resource$_T$; and can decrease capabilities, such as removing write access from an inherited read, write Resource$_T$.

Groups

Groups represent a set of login IDs. An individual Group$_T$ may be assigned one or more Roles$_T$. Individuals are assigned to Groups$_T$ by UserProfile membership.

3. Achieving User Identification

Individual (Human) Identification

The concept of users is described in RBAC$_0$. The effectiveness of access control rests on proper user identification [SS94]. The requirement to uniquely identify system users is not new. Those interviewed for this effort reported that the identifying information for individuals associated to system login accounts is collected and maintained by access request forms. Typically, the system administrator who actually implements the login account maintains a file of the paper request forms. The effort involved in associating a login account to an individual depends entirely on the system administrator's filing skills. The discovery of relationships between login accounts on separate systems would take longer or may not be discovered at all, due to separate administration teams. Finally, routine

validation of the individuals' continuing need for access was rarely conducted because of the effort involved.

One interviewee described a server where all accounts were reviewed to determine if they were still active. Initiated as a part of a server upgrade, nearly 15,000 of more than 50,000 accounts were determined to be "dead accounts" that had not been accessed recently. Of the remaining 35,000 accounts, while they could say that the accounts were active, they were less certain that continued access was still required or authorized. As an interviewee put it "When someone new comes in, they're all over us to create the account and get the user up and running. But, as far as we can tell, nobody every quits or transfers, since the customers rarely let's us know about users who leave or the customer only closes the basic login account and forgets about additional accesses that have been accumulated over the person's tenure."

Security Index

The project is required to provide centralized user correlation. To achieve this a Security Index will be implemented that supports the following concepts:

- $RBAC_0$ describes differentiation between users and sessions. An individual's login ID can be more closely related to a session, in that the act of logging onto a system is initiated by the individual as needed and invokes only those permissions associated to the active login ID. In larger organizations, a single user will typically have multiple login IDs. These login IDs, in all probability, will not be identical. The Security Index provides the ability to associate multiple login IDs to a single individual. This supports the ability to distinguish between users and sessions in the managed environment.
- The Security Index supports the requirement that "there must be a one-to-one correspondence between user identifiers and human beings"[San97] as fundamental to enforcement of $RBAC_2$ described constraints like exclusivity and cardinality.
- The Security Index establishes an integration point to pro-actively identify changes to the individual's characteristics. In most cases, the most accurate and most current information about an individual's status is held by the personnel office or management structure. Future phases of the project will integrate the ability to validate Security Index individual attributers against external information resources such as HR databases or organization directory servers. Validating against these external information resources will enable confirmation of the individual's continuing eligibility for role memberships; pro-active response to changes in the individual's qualifications; and, optionally, the automated suspension of affected role memberships.
- The Security Index will enable timely and complete suspension and subsequent removal of resource access in response to security events.
- Finally, selected constraints (roles) will be implemented as Security Index individual attributes, rather than roles. This is discussed later in section 4.

Mechanisms

The TME does not provide an explicit structure for user correlation. For this effort, a $UserProfile_T$, managed by the security office, is used to implement a Security Index of all individuals associated to the TMR. The Security Index could best be described as a role called "known individual", and membership a prerequisite to membership in any other regular or administrative roles described by the TME. A "UP" subscript is used to indicate discussion of the technical implementation, the UserProfile called Security Index, vs. the Security Index as a role.

Achieving User Identification

The $Security\ Index_{UP}$ User Name data element is the primary index key to uniquely identify an individual throughout the TME. This field is calculated from the individual's first name, middle name, last name and generational qualifier according to existing enterprise standards.

The $Security\ Index_{UP}$ Social Security Number is the secondary index key to uniquely identify an individual throughout the TME. The SSN is presented as clear text within the $Security\ Index_{UP}$, and as a cryptotext[1] field in all other UserProfiles to preserve the individual's privacy. Where an individual does not have a SSN, a passport number may be used or a unique, random number may be generated for this field.

Enforcing the Security Index Membership Constraint

As mentioned, membership in the Security Index is a prerequisite to membership in any other UserProfile. This is enforced by default and validation policies associated to all other UserProfiles.

Creation of new user records within any UserProfile is controlled by default policies that retrieve the individual's identifying information from the $Security\ Index_{UP}$ based on either the User Name or the SSN index keys. If the index key value cannot be located, creation of the user record is denied.

The $Tivoli\ User\ Locator_T$ utility provides an interface to view the set of User-Profile assignments for any given individual.

Once a user record is created, General category data elements, including User Name and SSN, cannot be edited within the

[1] A cryptotext field shields the contents with asterisks. A common technique for password entry fields.

UserProfile. These data elements can only be edited from within the Security Index$_{UP}$. Any Security Index$_{UP}$ updates are propagated to all corresponding UserProfile records.

In the event that the individual's Security Index$_{UP}$ record is modified all corresponding UserProfile records are validated and highlighted to the profile's Administrator$_T$ for deletion if the new status no longer meets the UserProfile constraints.

In the event that the individual's Security Index$_{UP}$ record is deleted, all corresponding UserProfile records are also highlighted to the profiles' Administrators$_T$ for deletion.

Deletion of UserProfile records associated to an individual, even if all profile records are deleted, cannot cause deletion of the Security Index$_{UP}$ record.

Security Index within Role Hierarchies

The Security Index does not participate in any role hierarchy described within the TME since it is neither senior to (doesn't confer membership in junior roles) nor junior to (doesn't inherit membership from senior roles) any other role descriptions within the TME.

Process Identification

The RBAC definitions of "user" may or may not include the concept of intelligent autonomous agents, such as processes that act independent of any individual. To achieve comprehensive security management, this effort must address all system accounts, including processes. Processes will be managed by Process Profiles. Process Profiles will provide a management interface for the processes associated with operating systems, application servers, and services. The concept of Process Profiles has the potential to generate a huge number of un-coordinated profiles. In hopes of avoiding this, "template" profiles will be provided for common operating systems, application servers and services. The ultimate goal is to link these profiles to the software signatures found in inventory products such as TME-10 Inventory.

Mechanisms

Again, the UserProfile will be used to establish Process Profiles. All UserProfiles created within the TMR are instantiations of the TivoliDefaultUserProfile object. This object is designed to maintain information about individuals with data elements for names, offices, and contact information. In order to maintain the integrity in the under-lying database design, a new instance of UserProfile, called the DefaultProcessProfile, is provided. The DefaultProcessProfile$_{UP}$ is used to instantiate Process Profiles.

DefaultProcessProfile

This profile will provide:

- Data elements specific to identifying processes, such as process name, version number, and key requirements.
- Default and validation policies appropriate to process identification
- The ability to express constraints associated with the managed processes. As an example, a process can require US Citizenship for access.

Template Process Profiles

Figure 2 provides examples of Template Process Profiles. Multiple process profiles may be applicable to any single managed device. For example, a Tivoli managed Windows NT endpoint that also provides an Oracle server would be subscribed to each of the profiles, Windows NT, Oracle, and Tivoli_EndPoint. The benefits achieved by this approach include:

- Centralized, consistent management of process related accounts, including enabling resource accesses and configuring specific system files and accesses.
- Centralized password control.
- Developing process understanding when designing Process Profiles will directly related to subsequent Security Manager Resource$_T$ design later in the project.

4. Alternate Implementation of Selected Constraints

RBAC$_2$ provides the idea of constraints to determine if values assigned to the various RBAC$_0$ components are valid and prevents assignment of non-valid values. The implementation of constraints is based on role memberships. As an example, a mutual exclusivity constraint requires the membership of an individual be limited to no more than one role from an identified set of roles. Among the various forms of constraints mentioned the constraint of prerequisite roles is pertinent to this portion of this paper. The concept of prerequisite roles is based on competency and appropriateness, whereby a user can be assigned to role A only if the user is already a member of role B. [San97].

The following constraints pertinent to the target environment are implemented as Security Index individual attributes instead of roles.

- Employment Status: { Corporate, Contractor, Customer, none, other }
- Employment Term: { Perm, Temp, Intermittent, none }
- Citizenship: { ISO Country Codes }
- SysAdmin Qualification Level: {1, 2, 3, none }
- Access: { Proprietary, Sensitive, Public, none }
- Access Review Date – date field

Figure 2. Example Process Profiles

Solaris2x	root, daemon, sys, bin, backup, smtp, adm, uucp, nuucp, listen, lp, noaccess, nobody
WindowsNT	Administrator, Guest
Tivoli_ManagedNode	Unix: tmesrvd WinNT: tmesrvd
Tivoli_EndPoint	Unix: lcfsrvd WinNT: lcfsrvd

The primary reason for implementing these as individual attributes is to facilitate validation against external data sources. Even so, Citizenship and Background Date present some interesting considerations.

Citizenship represents a large set of valid values that cannot be distilled into only a few. In practice, only a few of these values would be used within most environments. Implementing pre-requisite role constraints is straightforward. The constraint "must be a member of USA" probably would not be implemented unless the role USA already exists. Implementing exclusivity constraints becomes more difficult. As an example, a constraint "may not be a member of Restricted Countries" would require the creation of a number of roles for the solely to validate the constraint. A single exclusionary constraint can be handled easily by the complementary set. When multiple exclusions occur in an overlapping fashion, it is less clear how to implement this as proper "roles".

Access Review Date also presents a challenge, since this data element is specific to each individual. Date based constraints, such as "active license" (license not yet expired) or "Access Review within the last year" don't seem to lend themselves to the typical definition of a role, but do seem to be valid constraints.

5. Achieving ARBAC97 Capabilities

It is expected that the management environment will evolve over time, both from continued efforts by the project and as the security managers become more accustomed to role design. The goal is to enable security managers to safely delegate administrative duties, while ensuring that corporate and site policies are enforced. This issue of "decentralizing the details of RBAC administration without loosing central control over broad policy is a challenging goal for system designers and architects" [SBC+97]. In this area of Administrator$_T$ management, ARBAC97 (Administrative RBAC '97) is used as a point of reference. ARBAC97 provides a model for the administration of RBAC within the RBAC96 context using three components: user-role assignment (URA97), permission-role assignment (PRA97), and role-role assignment (RRA97).

Mechanisms

Tivoli Administrators

As described above, Administrators$_T$ describe mutually exclusive administrative roles for the TME. Like the Unix root account, the Root_Administrator is the senior most administrative role within the TME. The authority conferred with membership in all other Administrator$_T$ descriptions is based on the TME Administrative Roles$_T$ assigned to the Administrator$_T$ for each of the Policy Regions. Administrative Role$_T$ assignments are static and must be manually managed by either the Root_Administrator or an Administrator$_T$ possessing roles senior to those that are to be modified. Finally, administrative roles are independent and cannot be combined into hierarchies. These three factors -mutual exclusivity, static Administrative Roles$_T$, and the lack of a capability to create hierarchies- encourage a tendency to create a one-to-one relationship between individuals and administrative roles. This approach is only feasible in smaller environments where the security manager is personally familiar with or has immediate access to the administrative staff or in environments where administrators have very broad ranges of authority. As the environment becomes larger and more complex, this approach quickly becomes cumbersome; has the potential to become a subjective process; discourages the concept of least privileges; and doesn't support the ability to securely delegate administrative authority beyond the immediate level.

To overcome this issue, another specialized UserProfile is created. Now Administrators$_T$ are used much like Security Management Role$_T$ descriptions to combine a set of administrative capabilities into a single, assign-able administrative role. The Administrator Profiles are used like Security Management Groups$_T$ to assign groups of login IDs to one or more roles. Finally, a DNS slight of hand is used to take advantage of the distinction between the exclusivity constraint on login IDs instead of individuals. This approach permits a limited number of administrative role definitions that describe management capabilities to be defined and subsequently combined into administrative roles via the Administrator Profiles. An individual that requires multiple administrative roles, such as "Unix Auditor" and "Purchasing DBA" invokes the required roles by authenticating to the DNS hostname associated to the administrative role. TME auditing is maintained since the authenticated ID can be traced to the individual via the Security Index.

Policy Regions

Policy regions provide the ability to create a hierarchy with respect to administrative authority. Policy Regions bound Administrator$_T$ authority. Authority (permissions) is granted by assigning Administrator Roles$_T$ to the Administrator$_T$ for each Policy Region.

The ACL Editor

The Roles mentioned above are implemented in Access Control Lists for the actual methods that enable administration activities. As delivered, the TME does not provide a way to edit these ACLs. Because the delivered configuration presented a number of security concerns, some of which are highlighted below, the ACL Editor Utility is considered a requirement for this effort. The ACL Editor Utility enables the security manager to configure ACLs in a way that supports the secure delegation of selected administrative functions. Figure 3 provides a list of the Roles, the default ACLs and the recommended ACLs pertinent to this discussion.

Figure 3

Administrative Function	**Default ACL Assignment**			**Recommended ACL Assignment** ***(note the addition of four new Administrator Roles)***						
	Senior	Admin	Sec Admin	Senior	Admin	Sec Admin	Sec_ARA	Sec_GRA	SecMgr	AsstSecMgr
Add User		X			X				X	X
Edit User		X		X	X				X	X
Delete User		X		X					X	X
Add Sec Resource		X	X			X				X
Edit Sec Resource		X	X			X	X		X	X
Del Sec Resource		X	X				X		X	
Add Sec Role		X	X			X				X
Edit Sec Role		X	X			X	X		X	X
Del Sec Role		X	X				X		X	
Add Sec Group		X	X	X		X				X
Edit Sec Group		X	X	X		X		X	X	X
Del Sec Group		X	X					X	X	

ARBAC97 Features

User Role Assignment (URA97)

URA97 describes the administration of user-role assignments, specifically those issues related to management of administrative roles that have authority to affect the user-role assignments as described in RBAC96. Within the TME, this can be stated as the management of Administrators$_T$ with the authority to create, edit, and delete records within UserProfiles.

As delivered, the authority to create, edit, and delete records within the UserProfiles contained by a Policy Region are all conveyed by the Administrator Role$_T$ "admin". This means that any Administrator$_T$ with the ability to create a user account, also has the ability to edit and delete the record. This contradicts ARBAC97 concept of independent assignment of the administrative authorities *can-assign* and *can-revoke*. Also, this is not always a desirable situation. The creation of a user account only requires a validated requirement and can be performed by a person with limited skill sets using the UserProfile. On the other hand, deletion of user accounts may require additional activities to ensure that all associated files are archived and deleted. From a security perspective, separation of create and delete authority increases the barrier for unauthorized activities by the requirement for collusion. To correct this issue the ACL Editor makes it possible to alter the ACL lists for these activities. Figure 3 shows the default ACLs and recommended ACLs associated with key management activities.

Permission Role Assignment (PRA97)

PRA97 describes the administration of permission-role assignments, specifically those issues related to management of administrative roles that have authority to affect the permission-role assignment relationships as described in RBAC96. Within the TME environment, this can be stated as management of Administrators$_T$ with the authority to create, edit, and delete UserProfiles, Security Management Groups$_T$, Security Management Resources$_T$, and Security Management Roles$_T$. Also included in this issue is the ability to edit the corresponding default policies.

Again, as delivered, these capabilities are assigned to TMR roles in a way that combines the *can-assign* and *can-revoke* authorities. Figure 3 provides a set of recommended ACLs to alleviate this issue.

Role-Role Assignment (RRA97)

RRA97 describes the administration of role-role assignment, specifically those issues related to the management of administrative roles with authority to affect role-role relationships. RRA97 describes three fundamental types of roles. Ability Roles describe abilities (sets of permissions required to convey a specific ability) and may only have permissions or other Abilities Roles as members. Group Roles describe groups of individuals and may have users or other groups and members. Finally, a type of hybrid role called a UP-Role can contain any of users, permissions, Abilities Roles, Group Roles, or other UP-Roles.

Ability-Role Assignment (ARA97)

ARA97 describes administrative authority that would typically be delegated to staff members closest to the technology of the managed resource. This can include application administrators, integrators, and even development staff. In many instances combinations of staff members will be required to develop a well-defined Ability-Role. Within the TME, Ability-Roles are achieved by Resource Records$_T$ and subsequent Role$_T$ assignments. The resulting Roles$_T$ would describe discrete organizational activities that can be expressed by verb-noun combinations like "release web content", "authorize contract expenditure", and "assign cubicle".

The authority to create, edit, and delete Security Management records, including Resources, Roles, and Groups, is conveyed by the TME Administrator Roles "security_admin" or "admin". This combines both ARA and GRA into a single control mechanism so that an Administrator$_T$ assigned ARA authorities also will possess GRA authorities.

Group-Role Assignment (GRA97)

GRA97 describes administrative authority that could typically be delegated to personnel and team managers. Within the TME, GroupsT and the subsequent assignment to RolesT achieve Group-Roles. The resulting groups would describe organizational affiliations, units, and roles. Examples include "on-line customer", "intern", "Emergency Response Team", "Help Desk Consultant", and "Client Manager". Issues associated to authority to create, edit, and delete the Security Management records associated with GRA and a solution is discussed above.

6. Migration Tool.

Migration to new management environment will be accomplished as an incremental process over a period of time. To aid this effort, a migration tool will be provided to assist in analyzing the existing environment and migrating systems into the management environment. Required characteristics of this tool include:

- Support migration over a period of time. The tool cannot take a "now or never" approach. It must be able to incrementally process a target system into the managed environment in a manner that permits joint management by both traditional means and the management environment over a period of time.
- Actions must be 100% Reversible. Each incremental action taken by the migration tool must be able to be "backed out" in the event that the action creates an operational problem.
- No user-perceivable operational impact. Migration actions must be "behind the scenes" and without end-user perceivable impact to operational availability.
- Complete audit. The migration tool must maintain the integrity of the existing audit environment as well as provide audit of migration activities themselves.
- Encourage consistent role design. While User Administration provides the ability to automate user account management, it does not inherently encourage a RBAC style design. Migration capabilities provided with the product, namely the "populate" function, encourage the direct replication of the existing environment into the TME, resulting in a one-to-one correlation between managed system and UserProfiles. While this automates management of the environment, it also automates the problems associated with management of multiple systems on an individual basis.

RBAC Environment Analysis

While RBAC could be achieved with relative simplicity in a newly established environment, the migration of an existing environment presents significant challenges in how to analyze the environment and extrapolate roles that reflect both organizational function and support security policy enforcement. The Migration Tool is designed to aid this process by helping analyze systems to extract the groups, user accounts, process accounts, and administrator roles. The desire is to achieve a tool that permits organic analysis of the existing environment where the role designer can begin each session at any point in the migration process. As the session progresses, the tool will draw on only those issues that are required and permit the role designer to spin off a separate analysis branch to be returned to and followed up on later.

Reference Repository

A reference repository consisting of a series of configuration and hints files will be used to enable the basic tool to incorporate new service descriptions as the project progresses. Configuration files will control the basic physical execution of the migration tool. Hints files provide user configurable starting points for the various analysis modules within the migration tool. As an example, process hints files will provide

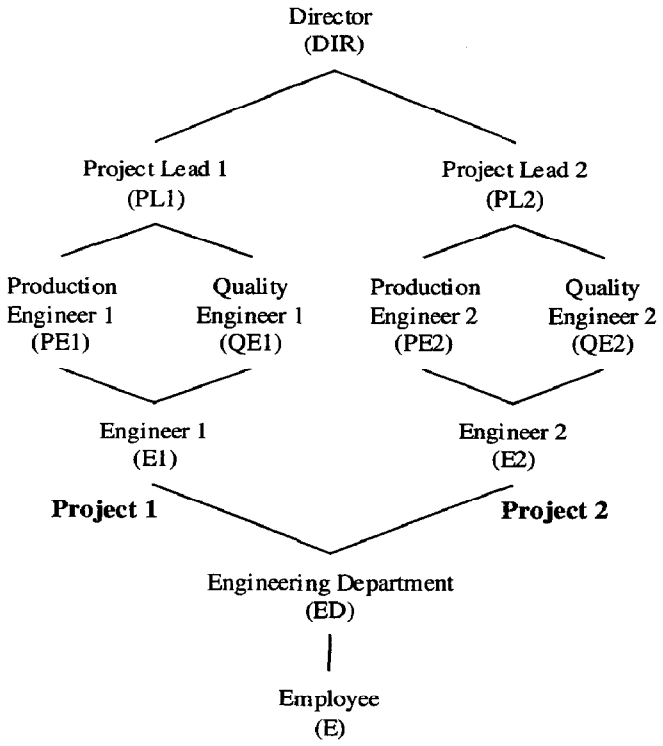

Figure 4. An example group hierarchy

the "typical" configuration for accounts used for "known processes" for each supported OS and common servers and applications. The role designer will be able to add to or modify these descriptions to make adapt the tool to the local environment.

Eventually, this may evolve to a meta-data source defining interfaces to various pertinent information resources. Examples of types of information resources that might be included are: organizational charts, personnel information, systems configuration management, network management, and CASE products such as Entity-Relationship-Diagrams, Data Dictionaries, Logical and Physical Schemas, Relationship Matrices, Event Analysis for Data Sources, Data Flow Diagrams, Connectivity Diagrams, and Systems Flow Charts.

Scope

The migration tool can only be used for devices managed by the TME. For Unix and WinNT operating systems, this includes ManagedNodes and EndPoints. At the start of a new migration session, the role designer will identify managed devices to be targeted during the design session. When a device has been identified to the management environment for the first time, the device is surveyed, key attributes logged into a central index, and the managed device identified as "in-progress". On each subsequent analysis, the indexed information is synchronized against the managed device and updated if required. At the point that the managed device becomes completely migrated to management environment, local management capabilities are terminated and the device identified as "wholly-managed". If local management capabilities are not terminated, then the system remains in-progress and is validated prior to each analysis session.

Unix or Windows NT Device Analysis

The migration goal is to identify roles that will cross device boundaries, but even where multiple devices are identified for a migration session, each device will have to be independently examined before they can be jointly analyzed. The remainder of this section discusses the evaluation of a single device and highlights where inter-device conflicts can be identified:

Device Survey

The first step is the initial device survey. This is accomplished using the Tivoli "populate" function to essentially slurp in all login account and group information for the device into a temporary UserProfile[2]. This holding profile is then used as the data source for all subsequent analysis of the device. The login account information is then processed as follows:

Processes Profiles

The goal is to manage system, applications, and process login accounts for all managed devices under a limited number of Process Profiles. By combining process accounts under a limited number of profiles, many management functions are also simplified. In the target environment, process account passwords must be changed on a routine basis. Process Profiles allow all account passwords (such as root or administrator) for the devices managed under a single profile to be changed with one management action. Known processes are those contained in the process hints file for each operating system and applications type. Potential accounts will be tested to determine if they are 100% consistent to the appropriate master process profile, if found consistent, the system is subscribed to the master. If not 100% consistent, then the hints file will be used to determine if the exception processes can be made consistent. As an example, if the exception is that the GCOS field does not match, typically this can be changed with no ill effect on the managed system, so this would be corrected and the system subscribed. In other cases, where exceptions involve issues such as group memberships, home directories, UIDs, or GUIDs, the hints file will be used to identify if these attributes can be automatically normalized without ill effect on

[2] Tivoli User Administration provides a GroupProfile that can be used to collect group information from Unix devices. Consideration was given to using the GroupProfiles for the migration process but it was decided that greater processing consistency would be achieved by not using them because (a) they do not support Windows NT and (b) group information is also collected into the individual user records.

the target system. If possible, the process is made consistent, then the device is subscribed to the master profile. If the process cannot be made consistent, then an exception profile created for the system.

User Profiles

Remaining accounts are then evaluated for potential user account groups. A hints file will provide terms for each field to look for in attempting to identify user accounts. As an example, key terms for the home directory attribute would include "home" and "guest". Of those interviewed, organizations generally have established processes for the identification of system users and construction of login names. This fact can be used by the role designer to update the hints file to make the process more accurate. As an example, at a site where all guest accounts start with GU_, the key term "^GU_" would be added to the hints file.

Identified accounts would then be distilled into potential profiles. A potential profile is a group of accounts identified as significantly similar. A group of accounts are considered significantly similar when after removing any comment field, such as the Unix GCOS field, and any string matching the account name they are equal.

The potential profiles are then presented to the role designer who must associate profile records to Security Index records. Once associated to the Security Index, the potential profile is processed again and split apart based on organization affiliations such as location or department. The resulting profiles are then generated with default and validation policies that generate fixed or calculated values for most attributes.

Next, we attempt to force significantly similar groups by removing attributes with the potential to be normalized, such as Windows NT remote connection drive, then repeating the process above.

The remaining accounts after this are then processed individually to determine if they are exceptional user accounts or process accounts.

The UserProfiles will most likely represent rudimentary role descriptions of organizational assignments. Figure 4 presents an example environment commonly used in RBAC discussions. Figure 5 demonstrates how UserProfiles can be combined through subscriptions to support the example environment OS level user access across multiple devices. You can see that the UserProfiles begin to model the organizational structure.

Inter-Device Management Issues

A significant issue when scaling to inter-device profiles is Unix UIDs. Many times, the same login name is used to represent an individual on all servers. However, interviewees reported that if there were similarities at the level of UIDs, it would be purely by chance. In attempting to centrally manage multiple

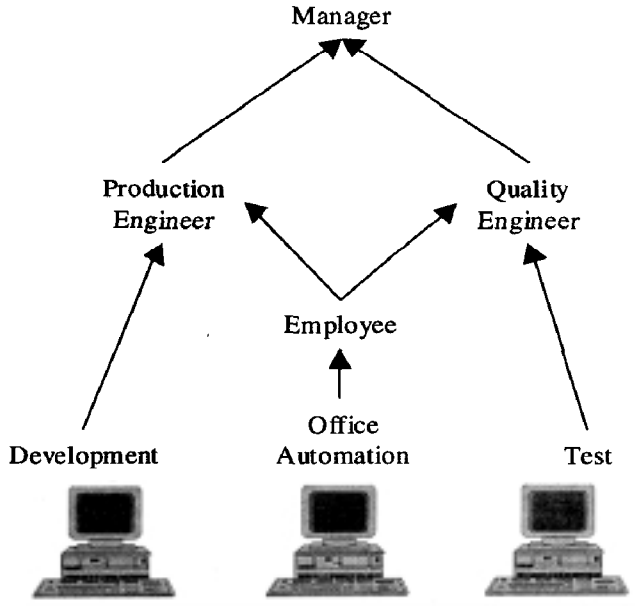

Figure 5. UserProfile Implementation

servers, there is a vast difference between systems that are managed in substantially the same manner and systems that are managed in *exactly* the same manner. The distinction between substantially the same and *exactly* the same is significant to the number of UserProfiles that will be required to manage user accounts.

Group Analysis

Traditionally, groups have been designed and implemented according to access requirements dictated by applications, such as the "Informix-Admin" group created by Informix Database Servers or the "Administrators" group in Windows NT and "sysadmin", (GID 14) on Solaris. The driving force for the creation of groups has been technology. Of the administrators interviewed, none reported the creation of groups for the purpose of identifying either user groups, such as a subordinate organization or customer, or user roles, such as a legal department or personnel department manager. Implementation of RBAC makes group management more significant. As a part of the account analysis process, existing groups have been decomposed to reflect organizational roles based on Security Index elements. But the Migration Tool will also allow the role designer to start with designing a group, assigning the appropriate roles to it, and subsequently assigning members from the Security Index to the group.

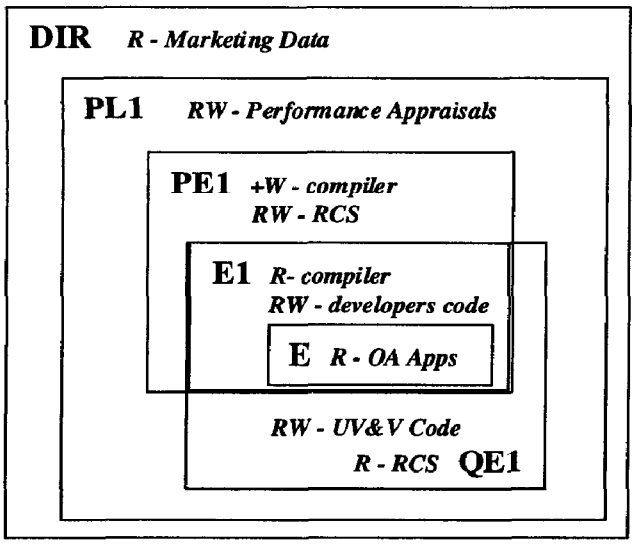

Figure 6. Resource-Role Implementation

Resource and Role Design

These two issues are handled together and will most likely be handled by the applications and systems administrator closest to the technology providing the resources. Resources and Roles are identified within SecurityProfiles provided by Security Manager. Modeling the environment into roles will be an iterative process. As Resources are identified and combined into roles, they can then be assigned to UserProfiles through Group assignments. Due to time constraints, the complexities of designing security profiles have not been fully explored and the manner in which the Migration Tool might help this process is not yet developed at this writing. Figure 6 provides a rudimentary view of how roles can be combined into a hierarchy to convey permissions and add unique permissions added to each role.

7. Conclusions

In this paper we've described initial efforts to achieve a GUI based security management interface for large complex information systems environments using the TME. Initial analysis indicates many RBAC concepts can be achieved with the TME, due in large part to the ability to customize the TME itself. The solution presented is not ideal in that some of the roles achieved are not readily obvious because they have been implemented as attributes of the individual and administrative hierarchies are obscured by the variety of mechanisms used to achieve them. The Security Index introduces a new level of user correlation, not typically found in large environments. We've shown that, with modifications, the TME can support ARBAC97 constructs that will enable secure delegation of administrative permissions. The migration tool provides an entry-level aid for migrating existing devices into the management environment and begins the process of associating resources to organizational functions. Much work remains on this project; continued integration of User Administration, incorporation of Security Management, broadening the supported operating systems, reaching up into the infrastructure layer, and continued enhancements to the migration tool. It remains to be seen if this initial offering will stand the test of implementation at the target sites, but the described solution is expected to provide a strong foundation for future integration efforts.

References

[San97] Ravi Sandhu, Role-Based Access Control, September 17, 1997

[SBC+97] Ravi Sandhu, Venkata Bhamidipati, Edward Coyne, Srinivas Ganata, and Charles Youman. The ARBAC97 Model for Role-Based Administration of Roles: Preliminary Description and Outline, *Proceedings of the 2nd ACM Workship on Role-Based Access Control*, ACM, 1997

[SS94] Ravi Sandhu and Pierangela Samarati, Access Control: Principles and Practice, *IEEE Communications, 32(9):40-48*, 1984

[TivA98] TME-10 User Administration, User and Group Management Guide, Version 3.6, Tivoli Systems, Inc, 1998

[TivF98] TME-10 Framework Planning and Installation Guide, Version 3.6, Tivoli Systems, Inc, 1998

[TivS98] TME-10 Security Management User's Guide, Version 3.6, Tivoli Systems, Inc, 1998