RBAC in the Solaris™ Operating Environment

White Paper

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
1 (800) 786.7638
1.512.434.1511

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Solaris Management Console, Solstice AdminSuite, and Trusted Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, Solaris Management Console, Solstice AdminSuite, et Trusted Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Please
Recycle

Contents

CHAPTER 1

Overview

In conventional UNIX® systems, root (also referred to as superuser) is all powerful, with the ability to read and write to any file, run all programs, and send kill signals to any process. In practical terms, this means that anyone who can become superuser has the power to modify a site's firewall, alter the audit trail, read through payroll and other confidential records, even bring down the entire network. It is no wonder that organizations no longer give out root passwords as freely as they used to.

Role-based access control (RBAC) is an alternative to the all-or-nothing superuser model. RBAC is in keeping with the security principle of least privilege, which states that no user should be given more privilege than necessary for performing that person's job. RBAC enables an organization to separate superuser capabilities and package them into special user accounts or *roles* for assignment to specific individuals according to their job needs. This enables a variety of security policies. Accounts can be set up for special-purpose administrators in such areas as security, networking, firewall, backups, and system operation. A site that prefers a single strong administrator but wants to let more sophisticated users fix portions of their own systems can set up an advanced-user role. As in many aspects of security, RBAC is not just a technology, it is a way of running a business. RBAC provides a means of reallocating system controls, but it is the organization that decides the implementation.

According to Joshi et al., in "Digital Government Security Infrastructure Design Challenges," *Computer Magazine*, February 2001, "Of the many technologies currently in development, RBAC models appear to be the most attractive solution for providing security features in a multidomain digital government infrastructure. RBAC features such as policy neutrality, principle of least privilege, and ease of management make them especially suitable candidates."

For a physical analogy illustrating the superuser model versus RBAC, consider a company where one pass key lets anyone into the building and all rooms are accessible. This is somewhat analogous to the superuser model: anyone with root password can do anything. If that company issues separate keys for utility areas such as the server room, network patch room, and boiler room, the situation is similar to an RBAC model (FIGURE 1). The employees responsible for these areas have separate keys according to their job duties.

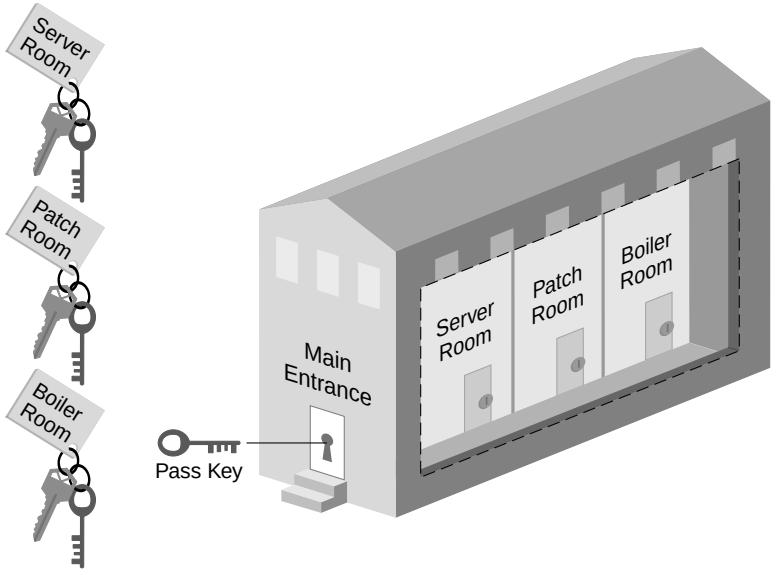

FIGURE 1 RBAC Multiple-Key Metaphor

CHAPTER 2

Solaris RBAC Implementation

The RBAC model in the Solaris Operating Environment is based on users logging in as themselves and assuming special identities that enable them to run restricted administration tools and utilities.

RBAC is fully compatible with Solaris auditing; the actions of a role are attributable to the user who assumed the role, and the audit records include the identity of the user, role, and effective ID used for policy overrides. The audit event mask of the user is augmented by that of the role.

The RBAC model introduces three elements to the Solaris Operating Environment:

- **Role**—A special identity that can be assumed by assigned users only.
- **Authorization**—A permission that can be assigned to a role or user to perform a class of actions otherwise prohibited by security policy.
- **Rights Profile**—A package that can be assigned to a role or user. It may consist of:

  - Authorizations

  - Commands with security attributes. The Solaris security attributes are the `setuid` functions for setting real or effective user IDs (UIDs) and group IDs (GIDs) on commands. Other systems, such as the Trusted Solaris environment, may use additional override mechanisms as security attributes.)

  - Supplementary (nested) rights profiles

FIGURE 2 shows how the RBAC elements fit in with the Solaris user administration. The arrows point from an attribute element to the element it can be assigned to. Solid arrows indicate preferred assignments. Dashed arrows indicate assignments that are possible but not considered secure.

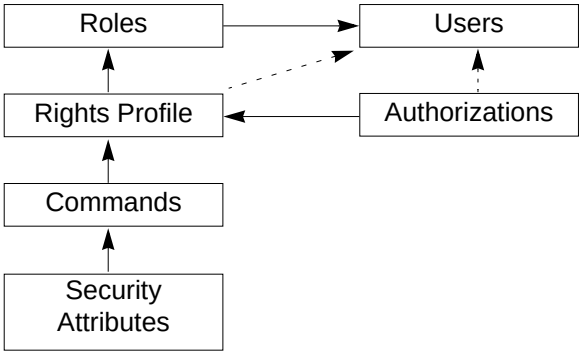

FIGURE 2 Solaris RBAC Element Assignments

Roles are assigned to users, enabling a user to assume a role. Rights profiles are assigned to roles, providing the root-like capabilities. Authorizations and commands with security attributes are components of rights profiles. Authorizations and rights profiles can be assigned directly to users, but this is not considered a secure practice.

Note – For those familiar with other RBAC models, the Solaris RBAC implementation has a flat structure for roles, and hierarchical structures for authorizations and rights profiles. A user can assume only one role at a time, and it must be assumed from the user's normal account. Authorizations are made hierarchical through the use of wildcards and a left-to-right naming convention. Rights profiles are made hierarchical through the ability to assign supplementary profiles to other profiles in a tree structure.

For those familiar with sudo, see "Appendix 2--Comparison of the RBAC Implementation with Sudo" on page 33."

Privileged Applications

Applications that override system controls by checking for authorizations or for specific UIDs or GIDs are considered to be *privileged applications*. These applications are not separate RBAC elements, but rather make use of authorizations and commands with security attributes.

The Solaris 8 environment, version 1/01, provides applications that check for authorizations:

- The entire Solaris Management Console suite of tools
- The batch job-related commands (`at`(1), `atq`(1), `batch`(1), and `crontab`(1))
- The device allocation commands (`allocate`(1M), `deallocate`(1M), and `list_devices`(1M))

By default, superuser is assigned all authorizations and thus can still execute these commands. The framework is now in place for sites that wish to delegate authorizations for using these applications.

Commands that must run with special IDs can be packaged with the needed security attributes in a rights profiles for assignment to users or roles.

Note – The preferred approach is to assign effective IDs, rather than real IDs, as security attributes to commands. Effective IDs are equivalent to the `setuid` functionality in the file permission bits and identify the user's ID for auditing. However, since shell scripts and other programs may require a real UID of root, real IDs must be available as well. For example, the `pkgadd` command requires a real, rather than effective, UID.

Authorized users can access privileged applications from the Solaris Management Console launcher or from the command line of a special shell called a *profile shell*. The profile shell is a Bourne, Korn, or C shell that has been modified to grant roles (and users) access to privileged commands assigned to their rights profiles.

Roles

A role is created in the same general manner as a user account, with a home directory, groups, password, and so on. Role information is stored in the `passwd`, `shadow`, `user_attr` (described later in this paper), and `audit_user` databases. All users who can assume the same role have the same role home directory, operate in the same environment, and have access to the same files. Users can assume roles from the command line by using `su` and supplying the role password. A user can also assume a role when opening a Solaris Management Console tool.

Users cannot login directly to a role; they must login to their user account first. There is no inheritance with roles; that is, when a user assumes a role, the role's attributes replace all of that user's attributes. The user's real UID is used for auditing purposes only. A user cannot assume a role directly while in a different role; roles are not hierarchical. A role's action can always be audited for the role ID and the user's real ID.

There are no predefined roles shipped with the Solaris 8 software, version 1/01. It is up to the customer site to decide what types of roles should be set up. However, there are three roles that can be readily configured by assigning one of the predefined rights profiles:

- **Primary Administrator** — For creating a role that can perform all administrative tasks, grant rights to others, and edit rights associated with administrative roles. It may also assign to others the Primary Administrator role and the ability to grant rights.
- **System Administrator** — For creating a role that can perform most nonsecurity administrative tasks. For example, the System Administrator can add new user accounts but cannot set passwords or grant rights to other users.
- **Operator** — For creating a role that can perform simple administrative tasks, such as backup, restore, and printer maintenance.

These rights profiles enable administrators to configure the suggested roles using a single profile, instead of having to mix and match rights profiles.

Sites that customize roles should pay close attention to the order of the rights profiles assigned to the role. The system does not prevent someone from entering multiple occurrences of the same command. The attributes assigned to the first occurrence of a command in a profile take precedence, and all subsequent occurrences are ignored.

Note – It is also possible to set up root as a role using a manual process. This prevents users from logging in directly as root, forcing them to log in as themselves first, and overrides the console setting in the /etc/default/login file. To make root a role, the root entry in the `user_attr`(4) file must be changed from `type=normal` to `type=role`. After this change is made, users can be assigned the root role through the Solaris Management Console launcher, the `rolemod`(1M) command, or by editing the user_attr database. See “Databases Supporting RBAC” on page 13 and “User Accounts Tool” on page 19 for more information.

Authorizations

An authorization is a discrete right granted to a user or role. RBAC-compliant applications can check a user’s authorizations prior to granting access to the application or specific operations within it. This is analogous to conventional UNIX applications checking for UID=0.

An authorization has a name that is used internally and in files (for example, *solaris.admin.usermgr.pswd*), and a short description that appears in the graphical interfaces (for example, `Change Passwords`). By convention, authorization names begin with the reverse order of the Internet name of the supplier followed by the subject area, any sub-area, and the function, all separated by dots, (for example,

com.xyzcorp.device.access). The exceptions to this convention are authorizations from Sun, which use the prefix *solaris* instead of an Internet name. This convention enables administrators to apply authorizations in a hierarchical fashion using a wildcard (*) to represent any strings to the right of a dot.

As an example of how authorizations are used, consider the following example of users in roles created using the predefined rights profiles. A user in the Operator role might be limited to the *solaris.admin.usermgr.read* authorization, which provides read but not write access to user configuration files. The System Administrator role naturally has both the *solaris.admin.usermgr.read* and the *solaris.admin.usermgr.write* authorizations for making changes to user files; but without the *solaris.admin.usermgr.pswd* authorization, the System Administrator cannot change a user's password. The Primary Administrator has all three of these authorizations. The *solaris.admin.usermgr.pswd* authorization is required to make password changes in the Solaris Management Console User Tool. It is also required for using the password modification options in the `smuser`(1M), `smmultiuser`(1M), and `smrole`(1M) commands.

An authorization that ends with the suffix *grant* permits a user or role to delegate any of their authorizations that begin with the same prefix. For example, a role with the authorizations *solaris.admin.usermgr.grant* and *solaris.admin.usermgr.read* can delegate the *solaris.admin.usermgr.read* authorization to another user. A role with the *solaris.admin.usermgr.grant* and *solaris.admin.usermgr.** can delegate any of the authorizations with the *solaris.admin.usermgr* prefix to other users.

TABLE 1 provides examples of how authorizations are used to limit command options in the Solaris environment.

TABLE 1

Command	Authorization Requirements
at(1), batch(1)	*solaris.jobs.user* required for all options (when `at.allow` and `at.deny` files do not exist)
atq(1)	*solaris.jobs.admin* required for all options
crontab(1)	*solaris.jobs.user* required for the option to submit a job (when `crontab.allow` and `crontab.deny` files do not exist); *solaris.jobs.admin* required for the options to list or modify other user crontab files

TABLE 1

Command	Authorization Requirements
allocate(1M) [if Basic Security Module (BSM) enabled only]	*solaris.device.allocate* (or other authorization as specified in `device_allocate` file) required to allocate device; *solaris.device.revoke* (or other authorization as specified in `device_allocate` file) required to allocate device to another user (-F option)
deallocate(1M) [if BSM-enabled only]	*solaris.device.allocate* (or other authorization as specified in `device_allocate` file) required to deallocate another user's device; *solaris.device.revoke* (or other authorization as specified in `device_allocate` file) required to force deallocation of the specified device (-F option) or all devices (-I option)
list_devices(1M) [if BSM-enabled only]	*solaris.device.revoke* required to list another user's devices (-U option)

Note – The authorizations currently available from Sun are stored in the `/etc/security/auth_attr` file. At this time, it is not possible to add new authorizations.

Rights Profiles

This section describes some typical rights profiles to demonstrate:

- The All rights profile provides a role access to commands without security attributes.
- The Primary Administrator, System Administrator, and Operator rights profiles are designed for specific roles. The Primary Administrator profile demonstrates the use of wildcards. The System Administrator profile uses discrete supplementary profiles to create a powerful role. The Operator profile uses a few discrete supplementary profiles to create a simple role.
- The Basic Solaris User rights profile shows how the `policy.conf` file can be used to assign tasks not related to security.
- The Printer Management rights profile is an example of a profile dedicated to a single area of administration.

The contents of the rights profiles are displayed in tables which label the purpose, authorizations, commands, supplementary rights profiles, and help files assigned. The help files are in HTML. They are stored in the directory `/usr/lib/help/profiles/locale/C` and can be readily customized if required. The Solaris Management Console Rights tool can also be used to inspect the contents of rights profiles.

All Rights Profile

The All rights profile uses the wildcard to include all commands with no security attributes. It is intended to provide a role access to all commands not explicitly assigned in other rights profiles. Without the All rights profile or some other rights profile that uses wildcards, a role has access only to explicitly assigned commands, which is not very practical.

Since commands in rights profiles are interpreted in the order in which they occur, any wildcard settings should be positioned last so that explicit attribute assignments are not inadvertently overridden. The All profile (if used) should be the final profile assigned.

TABLE 2

Rights Profile	Purpose / Contents
All	**Purpose**: Execute any command as the user or role. **Commands:** * **Help File:** RtAll.html

Rights Profiles for Specific Roles

The Primary Administrator, System Administrator, and Operator rights profiles are designed for specific roles. The Primary Administrator rights profile demonstrates the use of wildcards. The System Administrator rights profile demonstrates the use of supplementary rights profiles for a more powerful role. The Operator Role is an example of a rights profile with a more limited role.

Primary Administrator

The Primary Administrator rights profile is intended to be assigned to the most powerful role on the system, effectively providing that role with superuser capabilities. The *solaris.** authorization effectively assigns all of the authorizations provided by the Solaris software. The *solaris.grant* authorization lets a role assign any authorization to any rights profile, role, or user. The command assignment `*:uid=0;gid=0` provides the ability to run any command with UID=0 and GID=0. The help file `RtPriAdmin.html` is identified, so that a site can modify it if needed.

If the Primary Administrator rights profile is too powerful for a site's security policy, it can be modified or not assigned at all, provided that the security capabilities are handled in a different rights profile.

TABLE 3

Rights Profile	Purpose / Contents
Primary Administrator	**Purpose**: Can perform all administrative tasks. **Authorizations**: *solaris.*, solaris.grant* **Commands:** *:uid=0;gid=0 **Help File:** RtPriAdmin.html

System Administrator

The System Administrator rights profile is intended for the system administrator role. Since the system administrator does not have the broad powers of the primary administrator, no wildcards are used. Instead, discrete administrative rights profiles dealing with general administration are assigned. The rights profiles contains no authorizations associated with passwords, roles, or rights profiles. (The commands assigned to the supplementary rights profiles are not shown in this example.)

Notice that the All rights profile is assigned at the end of the list of supplementary rights profiles assigned to the System Administrator.

TABLE 4 s

Rights Profile	Purpose / Contents
System Administrator	**Purpose:** Can perform most nonsecurity administrative tasks. **Supplementary rights profiles:** Audit Review, Printer Management, Cron Management, Device Management, File System Management, Mail Management, Maintenance and Repair, Media Backup, Media Restore, Name Service Management, Network Management, Object Access Management, Process Management, Software Installation, User Management, All **Help File:** RtSysAdmin.html

Operator

The Operator profile is a less powerful administrative rights profile and provides the ability to do backups and printer maintenance. The ability to restore files has more security consequences, so the default is to not assign it to this rights profile.

TABLE 5

Rights Profile	Purpose / Contents
Operator	**Purpose:** Can perform simple administrative tasks. **Supplementary rights profiles:** Printer Management, Media Backup, All **Help File:** RtOperator.html

Basic Solaris User Rights Profile

By default, the Basic Solaris User rights profile is assigned to all users through the `policy.conf`(4) file. It provides basic authorizations useful in normal operation, typically read-only access to general resources and read-write access to the user's personal resources. Note that there is a trade off between the convenience offered by the Basic Solaris User rights profile and security. Those sites that need stricter security may prefer to remove this profile from the `policy.conf` file.

TABLE 6

Rights Profile	Purpose / Contents
Basic Solaris User	**Purpose:** Provides automatically assigned rights to all users. **Authorizations:** *solaris.profmgr.read, solaris.jobs.user, solaris.admin.usermgr.read, solaris.admin.logsvc.read, solaris.admin.fsmgr.read, solaris.admin.serialmgr.read, solaris.admin.diskmgr.read, solaris.admin.procmgr.user, solaris.compsys.read, solaris.admin.printer.read, solaris.admin.prodreg.read, solaris.admin.dcmgr.read* **Supplementary rights profiles:** All **Help File:** RtDefault.html

Printer Management Rights Profile

Printer Management is a typical rights profile intended for a specific task area. Both authorizations and commands are assigned to the Printer Management rights profile. (Note that only a partial list of commands is shown in the table).

TABLE 7

Rights Profile	Purpose / Contents
Printer Management	**Purpose:** Manage printers, daemons, and spooling. **Authorizations:** *solaris.admin.printer.delete, solaris.admin.printer.modify, solaris.admin.printer.read* **Commands:** /usr/sbin/accept:euid=lp, /usr/ucb/lpq:euid=0, /etc/init.d/lp:euid=0, /usr/bin/lpstat:euid=0, /usr/lib/lp/lpsched:uid=0, /usr/sbin/lpfilter:euid=lp, etc. **Help File:** RtPrntMngmnt.html

RBAC Example

FIGURE 3 illustrates how the elements interoperate to support the RBAC model.

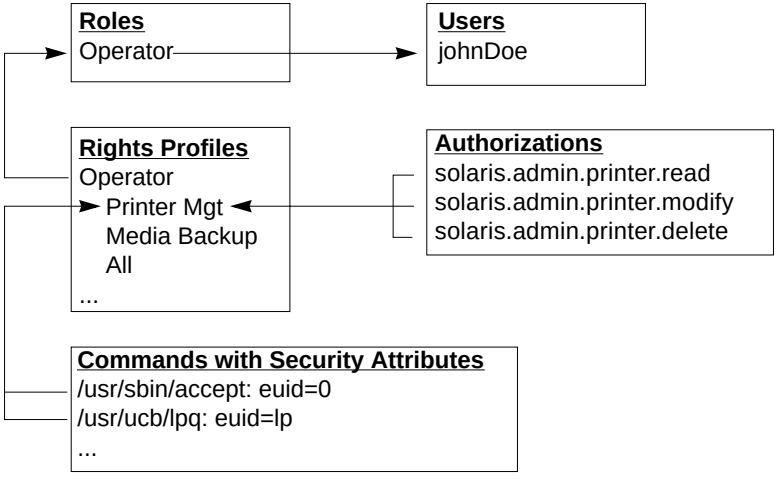

FIGURE 3 Solaris RBAC Element Assignment Example

The example has an Operator role for maintaining printers and performing media backup. The Operator role is assigned to user johnDoe, who can assume it by supplying the Operator password.

The Operator rights profile has been assigned to the Operator role. The profile has three supplementary profiles assigned to it. Printer Management and Media Backup reflect the role's chief tasks, and All is included for access to all other commands, but without any security attributes.

The Printer Management rights profile is for managing printers, print daemons, and spoolers. It has three authorizations: *solaris.admin.printer.read*, *solaris.admin.printer.modify*, and *solaris.admin.printer.delete* for manipulating information in the printer queue. The Printer Management profile also has a number of commands with security attributes assigned to it, `/usr/sbin/accept` with `euid=0`, and `/usr/ucb/lpq` with `euid=lp`, for example.

Databases Supporting RBAC

Data for the RBAC elements is stored in these four databases:

- `user_attr` (extended user attributes database) - Associates users and roles with authorizations and rights profiles.
- `auth_attr` (authorization attributes database) - Defines authorizations and their attributes, and identifies the associated help file.
- `prof_attr` (rights profile attributes database) - Defines profiles, lists the profile's assigned authorizations, and identifies the associated help file.
- `exec_attr` (profile execution attributes database) - Identifies the commands with security attributes assigned to specific rights profiles.

Note – The commands may also indicate a security policy. Currently, the only security policy available for the Solaris Operating Environment is `suser` (for superuser). The `suser` policy is the default; it accommodates both the ID attributes and authorizations. The Trusted Solaris environment, which can interoperate with the Solaris environment, uses a policy called `tsol`. Additional policies may be available in future releases.

The `policy.conf` database is also important to the RBAC implementation. It can contain authorizations and rights profiles to be applied to all users by default.

The `user_attr` database stores the basic definitions for both users and roles (they are differentiated by the `type` field). It contains the attributes shown in FIGURE 4, which includes a comma-separated list of profile names. The definitions of the rights

profiles are split between the `prof_attr` database, which contains profile identification information, authorizations assigned to the profile, and supplementary profiles, and the `exec_attr` database, which identifies the policy and contains commands with associated security attributes. The `auth_attr` database supplies authorization information to the Solaris Management Console tools. The `policy.conf` database supplies default authorizations and rights profiles to be applied to all users.

Each of these databases uses a `key=value` syntax for storing attributes. This approach accommodates future expansion of the databases, and allows a system to continue if it encounters a key unknown to its policy.

The *scope* of the RBAC databases can apply to individual hosts, or to all hosts served by a name service such as NIS, NIS+, or LDAP. The precedence of local configuration files versus distributed databases for the `user_attr` database is set by the precedence specified for the `passwd` entry in the file `/etc/nsswitch.conf`. The precedence for `prof_attr` and `auth_attr` are individually set in `/etc/nsswitch.conf`. The `exec_attr` file uses the same precedence as `prof_attr`. For example, if a command with security attributes is assigned to a profile that exists in two scopes, only the entry in the first scope is used.

These databases can be created manually or by using the `smattrpop`(1M) command. The databases can reside on a local system or they can be administered by the NIS, NIS+, or LDAP name service.

FIGURE 4 illustrates how the RBAC databases work together.

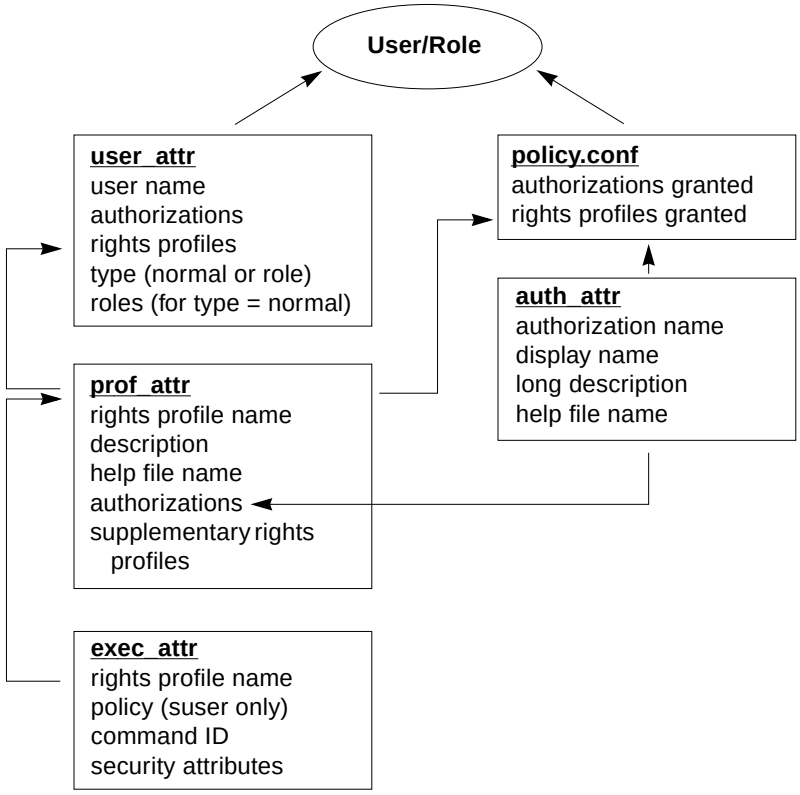

FIGURE 4 RBAC Database Relations

The databases can be edited manually or manipulated with the following commands:

- `smexec`(1M) — manage entries in the `exec_attr` database
- `smmultiuser`(1M) — manage bulk operations on user accounts
- `smuser`(1M) — manage user entries
- `smprofile`(1M) — manage profiles in the prof_attr and exec_attr databases
- `smrole`(1M) — manage roles and users in role accounts
- `useradd`(1M) — add new users
- `usermod`(1M) — change user files
- `rolemod`(1M) — change role files
- `roledel`(1M) — remove roles
- `roleadd`(1M) — add new roles

CHAPTER 3

Solaris Management Console Launcher

The Solaris Management Console launcher is a Java™ technology-based console for launching administrative tools. It is a key part of the RBAC implementation in the Solaris 8 environment, version 1/01.

However, the operation of the launcher and tools is beyond the scope of this paper. For more information, refer to the Solaris Management Console online help.

The Solaris Management Console tool suite interoperates with RBAC in four ways:

- Provides an interface for assuming roles and indicates role assumed in the tool windows
- Manages the elements of the RBAC infrastructure
- Restricts access to Solaris Management Console tools based on authorizations within the scope of the current server
- Executes legacy applications with security attributes

Main Window

When the launcher is first invoked, the main console window is displayed (FIGURE 5). At this point, there are no tools loaded and the default toolbox (*This Computer*) is displayed in the navigation pane.

The scope of a tool is specified in the tool box. It is either *files, nis, nisplus,* or *ldap.* Each of these is further qualified by the server name and domain.

Each toolbox has a defined scope of operation, that is, it can be set to use the RBAC databases on the local host or from the name service. The scope attributes govern the behavior of that toolbox and specify which users or roles are permitted to use it. In some cases, it may be useful to set up multiple toolboxes with different characteristics for different scopes.

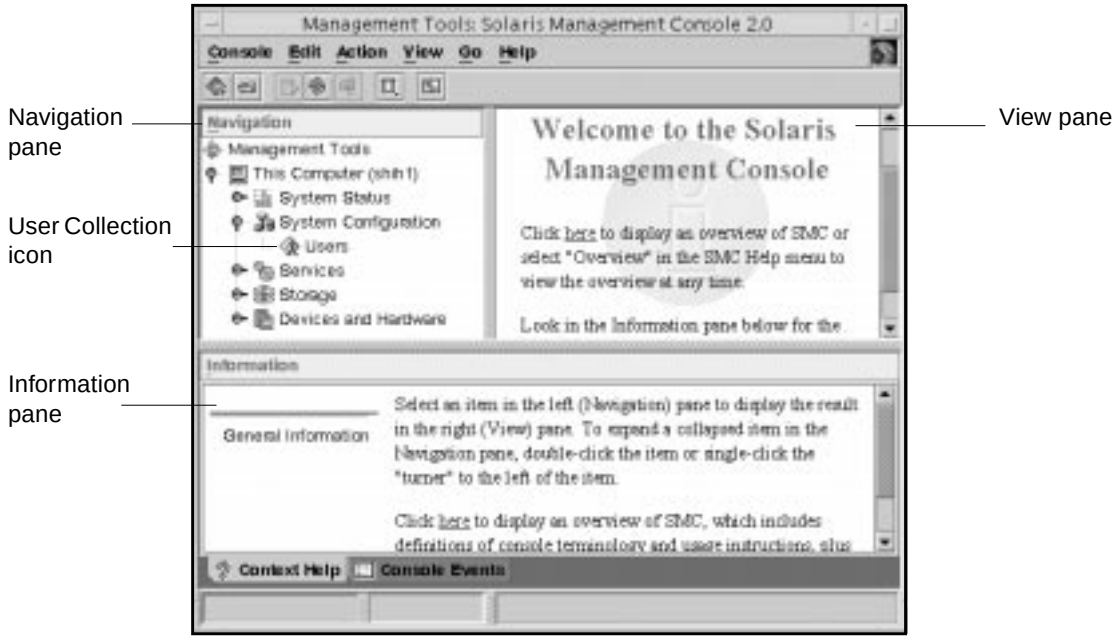

FIGURE 5 Initial State of the Main Solaris Management Console Window

This Computer operates on a local scope. It includes the following categories (folders) and tools, by default:

- System Status: Processes, Log Viewer
- System Configuration: Users
- Services: Scheduled Jobs, Solaris Management Console Server
- Storage: Mounts and Shares, Disks
- Devices and Hardware: Serial Ports

The next step is to select the toolbox that contains the appropriate tools and scope of operation. This can be accomplished by choosing New Console or Open Toolbox from the Console menu to select a different set of tools. Otherwise, simply double-click the tool or collection within the *This Computer* toolbox. RBAC management is handled by tools in the User Collection (FIGURE 4).

Assuming a Role through the Solaris Management Console Interface

Double-clicking a tool or collection displays a login dialog box for the user to be authenticated. If any roles are assigned to the user, a second login dialog box with a role option menu for assuming roles is displayed. The user should then assume the role with the capabilities appropriate to the tasks to be performed.

Managing RBAC Elements

The RBAC elements are managed by the following tools in the Solaris Management Console User Tool collection (FIGURE 6):

- User Accounts tool
- Administrative Roles tool
- Rights tool

Double-clicking a tool icon launches the tool and displays the data icons for that tool in the view pane at the right of the window.

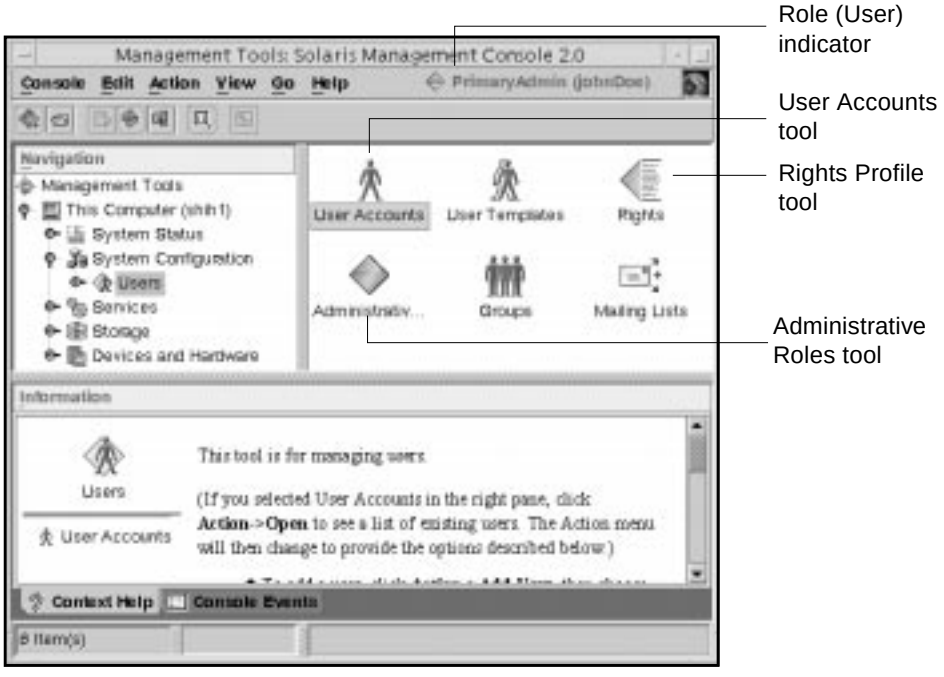

FIGURE 6 Solaris Management Console User Collection

User Accounts Tool

The User Accounts tool manages the rights profiles and roles assigned to a specific user (in addition to nonRBAC user data). Double-clicking a user icon displays the User Properties dialog box, for viewing or changing the current user's data. FIGURE 7 shows the User Properties dialog box with the Rights tab selected.

Rights profiles in the Available column on the left can be assigned to the current user; rights profiles in the Granted column on the right are already assigned to the user. The Add and Remove buttons are for switching rights profiles between columns. The Move Up and Move Down buttons change the order of the assigned rights profiles. As mentioned previously, the order of assignment determines which security attributes assigned to a duplicated command will take precedence.

The Roles tab operates in similar fashion for viewing or changing the roles assigned to the user.

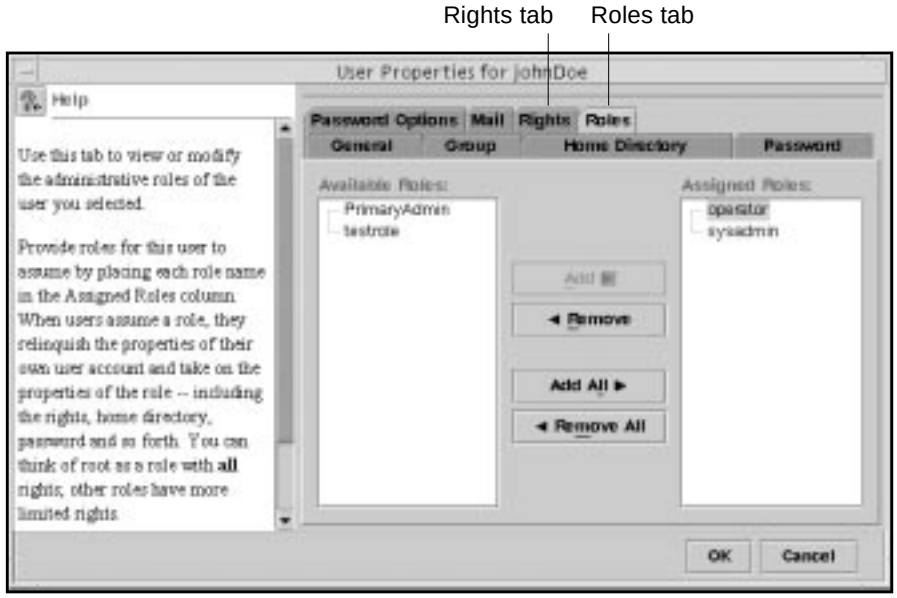

FIGURE 7 Assigning Rights Profiles in the User Properties Dialog Box

Administrative Roles Tool

The Administrative Roles tool is for defining a role and assigning users to the role. The properties dialog box for the Administrative Roles tool is similar to the User Tool version except for these minor differences:

- The Password Options, Mail, and Roles tabs in the User Tool are not available in the Roles Tool, because they are not applicable to roles.
- The Users tab in the Roles Tool provides a convenient means of assigning the role to users.
- The General tab for roles properties permits a profile shell (referred to as role shell in the GUI) to be selected for the role (FIGURE 8).

FIGURE 8 Administrative Roles Tool Properties Dialog Box

Rights Profile Tool

The Rights Profile tool is for building or modifying rights profiles, using commands with security attributes, authorizations, and supplementary rights profiles. FIGURE 9 shows how authorizations are assigned. Note that the authorizations are grouped to indicate their hierarchy.

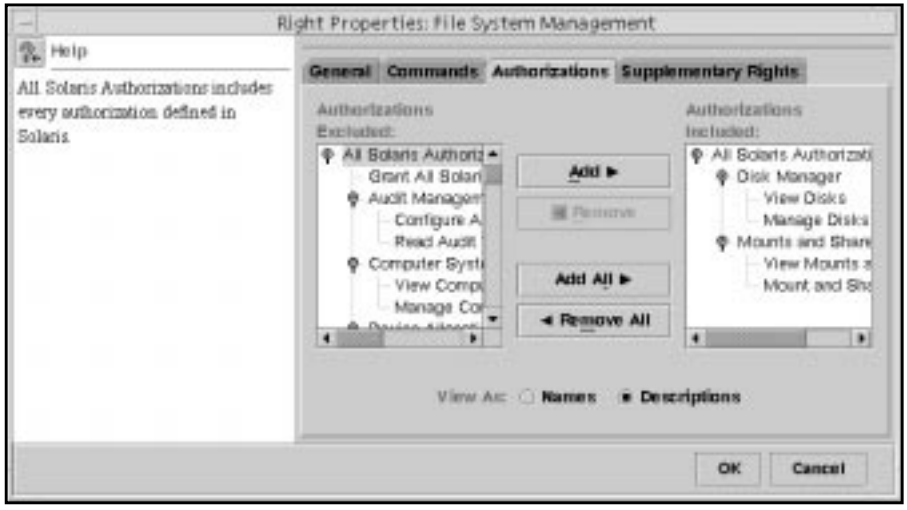

FIGURE 9 Assigning Authorizations in Rights Tool

Assignments of commands with security attributes are made in the Commands tab of the Rights tool. To add security attributes to a command, move the command to the Commands Permitted column, select it, and click the Set Security Attributes button at the bottom of the column. This displays a dialog box for choosing administrative user or group, and real or effective ID.

How Authorizations Restrict Solaris Management Console Operations

A site can use authorizations to control which roles can perform which tasks. TABLE 8 shows examples of how authorizations are required for specific operations.

TABLE 8

Tool	Dialog Box	Task (Required Authorizations)
User	Properties	View users (*solaris.admin.usermgr.read*) Edit users(*solaris.admin.usermgr.write*) Assign rights (*solaris.admin.profmgr.assign*) Delegate rights (*solaris.profmgr.delegate*) Assign roles (*solaris.role.assign*) Delegate roles (*solaris.role.delegate*) Change password (*solaris.admin.usermgr.pswd*)
Administrative Roles	Properties	View roles(*solaris.admin.usermgr.read*) Edit roles (*solaris.role.write*) Assign roles (*solaris.role.assign*) Delegate roles (*solaris.role.delegate*) Delegate rights (*solaris.profmgr.delegate*)
Administrative Roles	Add Role Wizard	Edit roles (*solaris.role.write*) Assign roles (*solaris.role.assign*)
Administrative Roles	Assign Role	Assign roles (*solaris.role.assign*) Delegate roles (*solaris.role.delegate*)
Right	Rights Properties	View rights (*solaris.admin.profmgr.read*) Edit Rights (*solaris.admin.profmgr.write*) Delegate rights (*solaris.profmgr.delegate*)
Rights	Add Right	Edit rights (*solaris.profmgr.write*)

Securing Legacy Applications

In addition to the default Solaris Management Console toolboxes, other tools and toolboxes can be made available to users. Custom tools can be JavaBeans™ applications that have been developed specifically for the Solaris Management Console launcher, other Java™ applications, or legacy applications. (The term *legacy application* refers to Solaris and Trusted Solaris applications that are not written specifically for the Solaris Management Console launcher.)

Sites are free to add their own Java and legacy applications to be accessed through the launcher. These applications can be restricted to specific roles. The Solaris Management Console interface automatically sets up X11 tools with the proper environment for remote display. Command-line applications can be started from the Solaris Management Console window. An additional terminal window is started for them automatically.

.

CHAPTER 4

Trusted Solaris RBAC Implementation

Sun's Trusted Solaris 8 Operating Environment is designed for deployments requiring enhanced security and policy enforcement. It uses the same model and databases to implement RBAC as the Solaris 8 environment. As mentioned earlier, the attributes in the databases are extensible; Trusted Solaris simply adds other key/value pairs. And the Trusted Solaris implementation has these additional features:

- CDE actions can have security attributes assigned and can be packaged in rights profiles.
- Two additional security attributes, *privileges* and *sensitivity labels*, can be assigned to commands and CDE actions.

A *privilege* is a discrete right granted to a process to perform an operation that would otherwise be prohibited by the Trusted Solaris environment. It is similar to an authorization but is assigned to processes rather than roles or users. Privileges can be passed from parent processes to the child processes they execute.

The `file_dac_read` privilege provides a good example of how privileges work. Processes cannot normally open data files unless they have the proper file permission. In the Trusted Solaris environment, the `file_dac_read` privilege gives a process the ability to override the UNIX file permissions for reading a file.

A *sensitivity label* is a tag applied to processes and files as part of mandatory access control. With mandatory access control, all users operate at a sensitivity label proportional to a level of trust, and all resources (files) are assigned sensitivity labels according to the degree to which specific classes of uses are permitted to see or modify them. The Trusted Solaris environment ensures that no process with an insufficient sensitivity label can access a file with higher sensitivity (at least not without an overriding authorization or privilege). Furthermore, no process can write a file at a lower sensitivity than the process's sensitivity label; this protects sensitive information from being downgraded.

FIGURE 10 compares the dialog boxes that apply security attributes in the Rights tool in both the Trusted Solaris and Solaris environments.

Solaris Set Security Attributes dialog box

Trusted Solaris Set Security Attributes dialog box

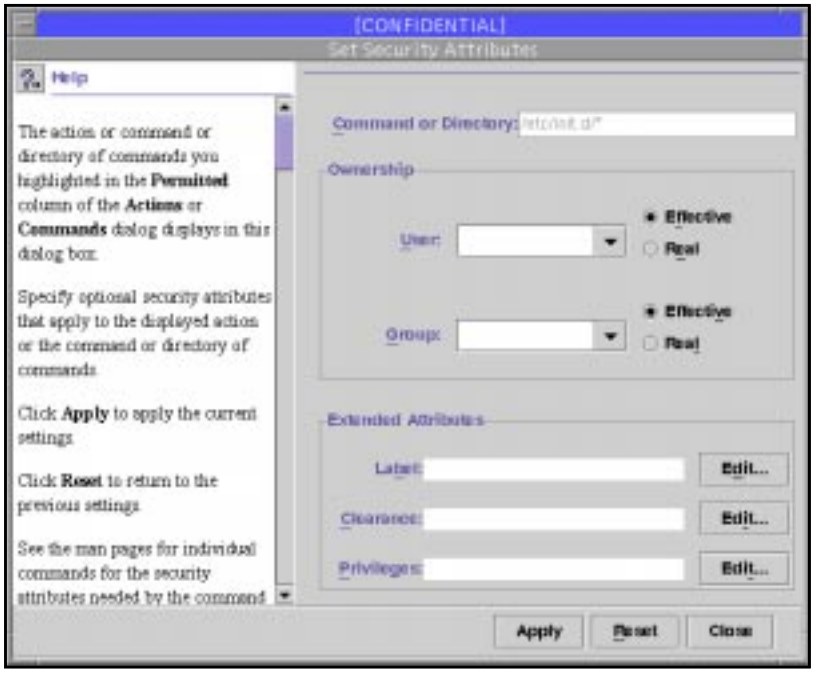

FIGURE 10 Comparison of Solaris and Trusted Solaris Security Attributes

FIGURE 11 shows all RBAC elements available in the Trusted Solaris environment; those RBAC elements unique to the Trusted Solaris Environment appear in shaded boxes.

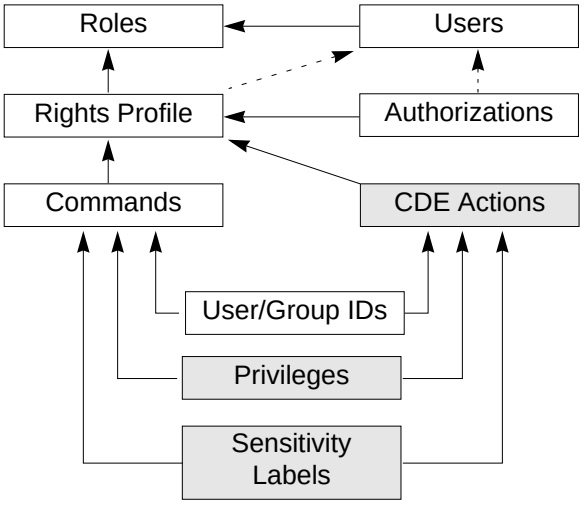

FIGURE 11 Trusted Solaris RBAC Element Assignments

The Solaris Management Console interface and the five RBAC databases (`user_attr`, `auth_attr`, `prof_attr`, `exec_attr`, and `policy.conf`) are used in the Trusted Solaris environment and are fully compatible with the Solaris environment. If a system running either the Trusted Solaris or Solaris environment encounters unrecognized attributes (key-value pairs) in these databases, the attributes are simply ignored. It is thus possible to administer hosts in one environment from a server in the other environment.

For more information on the Trusted Solaris environment, see the “Trusted Solaris 8 Operating Environment” white paper at www.sun.com/software/whitepapers.html#security.

CHAPTER 5

Appendix 1--RBAC Example Instructions

This appendix contains four procedures useful for configuring RBAC:

- *Starting the Solaris Management Console Tools*
- *Installing a Role*
- *Creating a New Rights Profile*
- *Adding a Rights Profile to a Role*

Starting the Solaris Management Console Tools

This section demonstrates how to run the User Tool collection.

1. **Start the launcher from the command line by typing `smc&` or by clicking the Solaris Management Console icon in the Tools subpanel in the CDE front panel.**

   The main window is displayed. No tools will be available until a toolbox has been loaded, and the user has been authenticated.

   If this is the first time using the Solaris Management Console program, click in the view pane for the online help overview. The online help provides context-sensitive help for both individual fields and general help topics.

2. **Navigate to the toolbox that covers the scope in which the change is to be made.**

   If the system is new, the only toolbox available is the one that covers the local server (*This Computer*). Otherwise, one of the toolboxes for the other scope may be selected. Toolboxes are defined using the toolbox editor.

3. **Click on the toolbox icon and System Configuration folder to open them, and then click on the User collection icon.**

   The authentication dialog box is displayed. It is required prior to the initial loading of all SMC tools and tool collections.

4. **Provide authentication and click the OK button.**

   To make changes when the roles are not yet installed, enter root and the root password. Root can run all the tools.

   If no roles are assigned to you, enter your name and password to get access to those tools permitted for normal users.

   If any roles are assigned to your account, select a role (or no role) and click Login with Role or Login without Role (FIGURE 12), and authenticate yourself in the dialog box that is displayed.

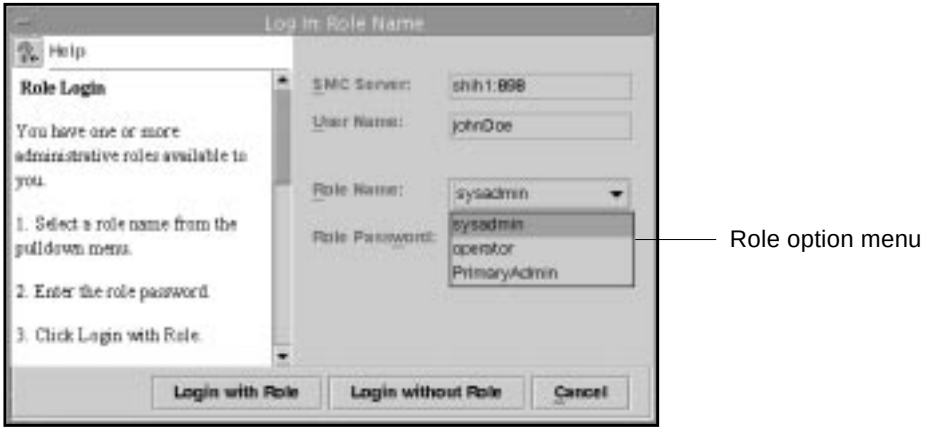

FIGURE 12 Role Login Dialog with Role Option Menu Displayed

Installing a Role

This section explains how to install a role. If the Primary Administrator role (or a similarly powerful role that can create other roles) is installed, assume this role, as shown in Step 4 on the previous page; otherwise enter authentication for root.

1. **After the user collection has been loaded, double-click the Administrative Roles icon.**

   The Administrative Roles icon (with the other user tools) is displayed in the view pane and also in the navigation pane (although it may be necessary to click the turner icon to the left of the Users icon to display the user tools).

2. **Select Add Administrative Role from the Actions menu.**

   This starts the Add Administrative Role wizard, a series of dialog boxes requesting information necessary for configuring a role. The first dialog box is "Step 1: Enter a role name." See FIGURE 13 on page 29.

3. **Enter the short version of the role name and the other identification information, and click Next.**

   Click the mailing list button to create an alias of users who can assume this role.

4. **In the "Step 2: Enter a role password" dialog box, enter the password in the Role Password field and again in the Confirm Password field. Then click Next.**

   Confirmation helps prevent a misspelled password from being saved.

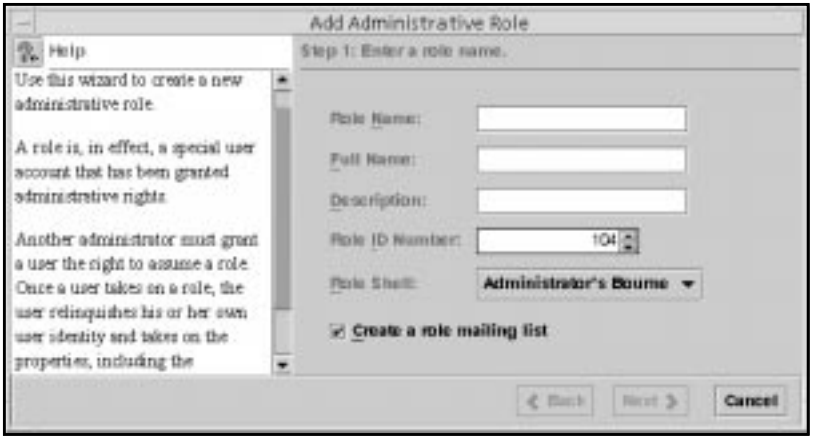

FIGURE 13 Add Administrative Role Wizard

5. **In the "Step 3: Enter role rights" dialog box, select the rights profiles to be assigned to this role. Then click Next.**

   This is done by double-clicking the desired rights profiles in the Available Rights column. The rights profiles in the Granted Rights columns are the ones that are assigned to this role. To configure the suggested roles, assign the rights profile of the same name to that role. All the necessary rights profile assignments have been prepackaged. For example,the Primary Administrator rights profile would be assigned to the Primary Administrator role.

6. **In the "Step 4: Select a home directory" dialog box, specify the server and path for the home directory. Then click Next.**

7. **In the "Step 5: Assign users to this role" dialog box, enter the login names for any users that are to be assigned to this role. Then click Next.**

Any users that are added must be defined in the same scope in which you are working.

If you selected the e-mail alias in the “Step 1: Enter a role name” dialog box, these users will receive e-mail addressed to the role.

8. **Check the information in the Review dialog box. Click the Finish button if the information is correct.**

   If there is missing or incorrect information, click the Back button successively to display the dialog box where the incorrect information is displayed.

Creating a New Rights Profile

This section demonstrates how to create a new rights profile. In this example, a rights profile named Restart is created. It sets EUID=0 for all commands in the `/etc/init.d` directory. This rights profile would be useful for roles like Operator or System Administrator that need to start and stop daemons.

1. **Click the Rights tool icon to begin the process of creating a new rights profile.**

   The Rights tool is loaded.

2. **Select Add Right from the Action menu.**

   The Add Right dialog box is displayed, set to the General tab.

3. **In the General tab, enter the name of the new profile “Restart,” the Description “For running initialization and termination scripts for daemons in the /etc/init.d directory,” and a help file name “Restart.html.”**

4. **Click the Commands tab.**

   The command assignment fields are displayed. The Commands Denied column (on the left) permits directories and commands to be selected for the rights profile. The Commands Permitted column (on the right) is empty at this point, because it is a new rights profile. The control buttons between the columns are for assigning or removing commands or changing the order of assigned commands. The Add Directory button allows a new directory to be added to the left column.

5. **Click the Add Directory button, enter the name of the directory “/etc/init.d” in the resulting dialog box, and click the OK button in the dialog box.**

   The `/etc/init.d` directory now appears in the left column.

6. **Select the `/etc/init.d` directory in the left column and click the Add button between the two columns.**

The `/etc/init.d` directory and its contents are moved to the right column and thus are assigned to the new rights profile. This includes any future commands added to the directory.

7. **With the `/etc/init.d` directory selected, click the Set Security Attributes button at the bottom of the right column.**

   The dialog box for setting UIDs and GIDs is displayed (FIGURE 14).

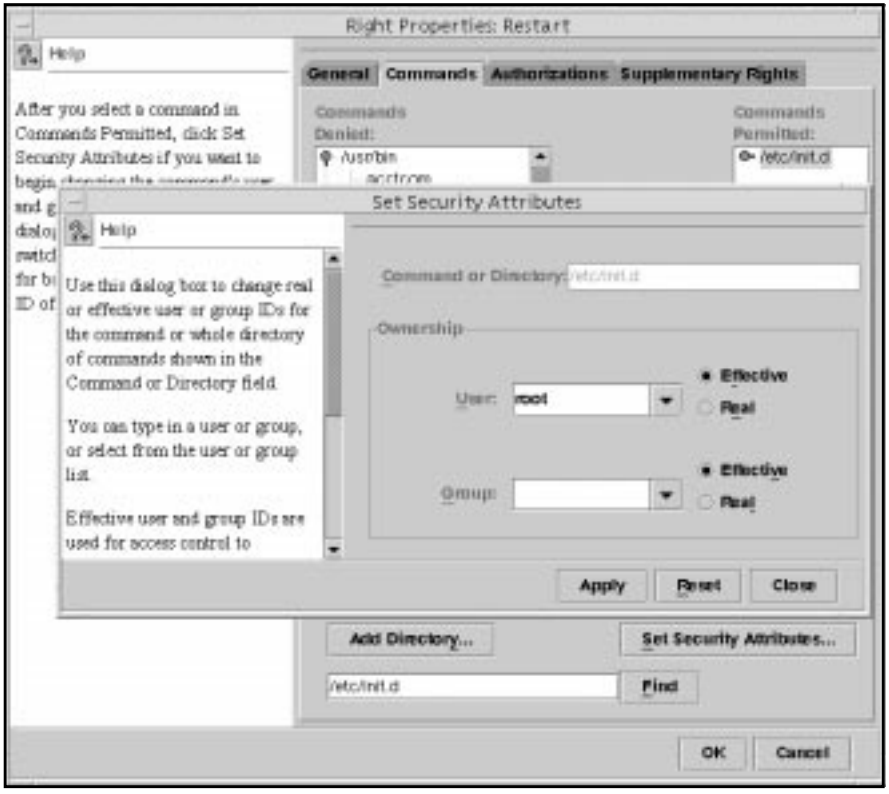

FIGURE 14 Setting Security Attributes

8. **With the Effective button selected for User Ownership, enter 0 in the User field, click the Apply button, and then the Close button.**

   The security attribute “EUID=0” is added to all commands in the /etc/init.d directory.

9. **Click OK at the bottom of the Add Right dialog box.**

   The dialog box is closed and the Restart rights profile is now available in the Rights tool.

Adding a Rights Profile to a Role

This section demonstrates how to add a rights profile to an existing role.

1. **Click the Administrative Roles icon in the navigation pane.**

   The Roles tool is loaded. It displays the existing roles in the View pane.

2. **Double-click the Sysadmin role (or some other test role).**

   The Roles Properties dialog for making changes to roles is displayed.

3. **Click the Rights tab in the dialog box.**

   The rights selection fields for assigning rights are displayed.

4. **Find and double-click the Restart rights profile in the Available Rights column.**

   The Restart rights profile is moved to the Granted Rights column, which assigns it to this role.

5. **Click OK in the Roles Properties dialog.**

   The assignment is saved.

   Thus, a new rights profile named Restart has been created, with all commands in the `/etc/init.d` directory set to EUID=0, and it has been assigned to an existing role.

CHAPTER 6

Appendix 2--Comparison of the RBAC Implementation with Sudo

There are some similarities between the sudo freeware package (offered by Todd Miller and Chris Jepeway) and the RBAC implementation.

The RBAC implementation uses roles in similar fashion to the sudo `User_alias`. The `User_alias` feature is used like conventional groups. Roles can have rights profiles, including authorizations and commands with security attributes, directly assigned to them. The roles require authentication prior to assumption.

Sudo uses `Runas_alias` to assign UIDs and GIDs. These assignments include real and effective IDs together. The RBAC implementation uses a finer-grained approach, so that either effective or real IDs can be assigned. Assigning an effective ID rather than a real ID enables the real user to be attributed for auditing purposes.

Sudo uses the `Host_alias` to provide host-specific controls. The RBAC implementation can provide host-specific controls by storing the RBAC databases on the local host, or an organization can use a name service to distribute the information.

The sudo `Cmd_alias` is similar to rights profiles in that it is a way to group commands.

In summary, sudo and the RBAC implementation accomplish the same basic objectives. The RBAC implementation has a GUI, a finer granularity, and name service compatibility. Most importantly, sudo is freeware, but RBAC is supported by Sun.

CHAPTER 7

Resources

Additional information about RBAC and the Solaris Operating Environment is available at www.sun.com. Specifically, these documents were consulted when writing this paper:

- "Solaris 8 System Administration Supplement," *Solaris 8 1/01 Update Collection*, Sun Microsystems, Inc., 2001, docs.sun.com.
- "System Administration Guide, Volume 2," *Solaris 8 System Administrator Collection*, Sun Microsystems, Inc., 2000, docs.sun.com.
- "Trusted Solaris Administration Overview," *Trusted Solaris 8 Answerbook*, Sun Microsystems, Inc., 2000, docs.sun.com.
- "Trusted Solaris 8 Operating Environment, A Technical Overview," Sun Microsystems, 2001, www.sun.com/software/whitepapers.html.

Additional Resources

These publications were also consulted when writing this paper:

- Joshi, James; Ghafoor, Ari; Aref, Walid G.; and Spafford, Eugene H., "Digital Government Security Infrastructure Design Challenges," *Computer Magazine*, February, 2001.
- Faden, Glenn, *RBAC in UNIX Administration*, Proceedings of the fourth Association for Computing Machinery (ACM) workshop on role-based access control, Pages 95 - 10, Sun Microsystems, 1999, www.acm.org/pubs/citations/proceedings/commsec/319171/p95-faden/.

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303

1 (800) 786.7638
1.512.434.1511

www.sun.com/solaris

Printed in the USA April 2001