



Trusted Solaris Administration Overview

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part Number 805-8119-10
December 2000

UNCLASSIFIED

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 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, docs.sun.com, AnswerBook, AnswerBook2, Trusted Solaris, and 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.

Federal Acquisitions: Commercial Software – Government Users Subject to Standard License Terms and Conditions

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 94303-4900 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 du système 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, docs.sun.com, AnswerBook, AnswerBook2, et 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.



Contents

Preface	11
1. Introduction to Administration	15
Basic Concepts Review	15
How the Trusted Solaris Environment Protects Against Intruders	16
How the Trusted Solaris Environment Enforces Access Control Policy	16
How the Trusted Solaris Environment Implements Administration	17
Understanding Trusted Software Administration	17
Overview of Trusted Software Administration	17
Understanding Roles	20
Understanding Rights Profiles	21
Understanding Authorizations	25
Understanding Privileges	29
Understanding Labels	31
Dominance Relationships Between Labels	32
Label Encodings Files	33
Label Ranges	33
How Labeled Files are Stored	35
Applying Labels to Email	37

- Applying Labels to Printed Output 38
- How the Trusted Solaris Environment Controls Device Access 39
 - Device Allocation 40
 - Device Label Ranges 40
 - Administering Devices through the Device Allocation Manager 41
- 2. Quick Tour of the Admin Tools 47**
 - Introduction to Trusted Solaris Administration 47
 - Accessing Tools in a Role Workspace 47
 - Remote Administration 48
 - Solaris Management Console Tools 48
 - Trusted CDE Actions 53
 - Administering Users 56
 - Default User Attributes 56
 - User Attribute Databases 57
 - Managing Users from the Command Line 59
 - Managing Users through the SMC 59
 - Administering Hosts and Networks 64
 - Security Families Tool Set 66
 - Administering Other Aspects of the Trusted Solaris Environment 68
 - File Management Commands 68
 - File System Management Commands 68
 - Mount Management 68
 - Process Commands 69
- 3. Administering Trusted Networking 71**
 - Overview of Trusted Solaris Networking 71
 - Homogeneous Networks 72
 - Heterogeneous Networks 72
 - Trusted Solaris Data Packets 73

- Security Families 74
- Related Subsystems 78
- Routing in Trusted Solaris 78
 - Loading Routing Information at Boot Time 78
 - Routing Tables in the Trusted Solaris Environment 78
 - Accreditation Checking 79
 - Routing Example 81
 - Using Routing Commands 81
 - Routing through Non-Trusted Solaris Gateway Clusters 82
- Trusted Solaris Network Commands 83
- Troubleshooting Networks 84
- Overview of Trusted NFS Mounting 85
 - Specifying Security Attributes for Mounting 86
- 4. Administering Auditing 87**
 - Planning and Setting Up Auditing 87
 - Audit Classes 87
 - Public Objects 88
 - Audit Information Storage 88
 - Audit Configuration Files 89
 - Auditing Tools 89
 - audit and auditd 89
 - auditconfig 90
 - audit_startup 90
 - audit_warn 90
 - praudit 90
 - auditreduce 91
 - auditstat 91
- Index 93**

Tables

TABLE P-1	Typographic Conventions	12
TABLE 1-1	Roles and their Responsibilities	20
TABLE 1-2	Rights Profile Descriptions	22
TABLE 1-3	Authorizations	26
TABLE 1-4	Privilege Categories	30
TABLE 1-5	Examples of Label Relationships	32
TABLE 1-6	Adornment—Related Commands	36
TABLE 2-1	Administrative Actions, Purposes, and Default Roles	53
TABLE 2-2	User Properties Summary	61
TABLE 2-3	Rights Manager Dialog Box Summary	63
TABLE 2-4	Template Dialog Box Summary	67
TABLE 3-1	Security Attributes by Host Type	76
TABLE 3-2	tnrhdb Fallback Mechanisms Example	77

Figures

Figure 1-1	Security Element Assignment in the Trusted Solaris Environment	19
Figure 1-2	Assigning Rights Profiles to Users	22
Figure 1-3	Rights Dialog Box	25
Figure 1-4	Normal Viewing of a Directory	37
Figure 1-5	Viewing the Contents of Multiple SLDs	37
Figure 1-6	Typical Print Banner Page	39
Figure 1-7	Device Allocation Administration Dialog Boxes	41
Figure 1-8	Device Administration Dialog Box	42
Figure 1-9	Device Allocation Configuration Dialog Box	43
Figure 1-10	Device Allocation Authorizations Dialog Box	44
Figure 2-1	Typical Trusted Solaris SMC	49
Figure 2-2	Simple SMC Tool Example	51
Figure 2-3	Tabbed SMC Tool Example	52
Figure 2-4	SMC Wizard Example	53
Figure 2-5	User Database Relationships	58
Figure 2-6	SMC User Tool Collection	59
Figure 2-7	User Properties Dialog Box	61
Figure 2-8	Rights Properties Dialog Box	63
Figure 2-9	Computers and Networks Tool Collection	65

Figure 2-10	Computer Properties Dialog Box	66
Figure 2-11	Modify Template Dialog Box	67
Figure 3-1	Homogeneous Network	72
Figure 3-2	Heterogeneous Network	73
Figure 3-3	Comparison of Data Packet Formats	73
Figure 3-4	Typical Trusted Solaris Routes and Routing Table	81
Figure 3-5	Tunneling Example	83

Preface

The *Trusted Solaris Administration Overview* is an introduction to administering the Trusted Solaris™ environment. As prerequisites, you should be familiar with basic system administration in the UNIX environment, understand security policy concepts, and should read the *Trusted Solaris User's Guide*.

Related Materials

The Trusted Solaris documentation set is supplemental to the Solaris 8 documentation set. You should obtain a copy of both sets for a complete understanding of Trusted Solaris. The Trusted Solaris documentation set consists of:

- *Trusted Solaris Documentation Roadmap* shows all volumes in the documentation set.
- *Trusted Solaris 7 Release Notes* presents information regarding the hardware requirements for installing Trusted Solaris, features included in the release, any known problems, and interoperability with previous versions.
- *Trusted Solaris Installation and Configuration* describes the process of planning for, installing, and configuring a new or upgraded Trusted Solaris system.
- *Trusted Solaris User's Guide* describes basic features of the Trusted Solaris environment from the end user's point of view.

Note - *Trusted Solaris User's Guide* contains a glossary that applies to the entire documentation set.

- *Trusted Solaris Administrator's Procedures* provides detailed information for performing specific administration tasks.

- *Trusted Solaris Audit Administration* describes the auditing system for system administrators.
- *Trusted Solaris Label Administration* provides information on specifying label components in the label encodings file.
- *Trusted Solaris Reference Manual* is a printed version of the man pages available in the Trusted Solaris environment.
- *Compartmented Mode Workstation Labeling: Encodings Format* describes the syntax used in the label encodings file for enforcing the various rules concerning well-formed labels for a system.
- *Trusted Solaris 7 Transition Guide* provides an overview of the differences between Trusted Solaris 1.x and Trusted Solaris 2.5.

How This Guide is Organized

Chapter 1 provides an overview of basic concepts needed to administer Trusted Solaris.

Chapter 2 presents an overview of the tools available in the Trusted Solaris environment, how they are accessed, and the databases on which they operate.

Chapter 3 provides an overview of how networking is implemented in the Trusted Solaris environment and discusses the tools for administering networking.

Chapter 4 describes the basics of performing auditing in the Trusted Solaris environment.

Typographic Changes and Symbols

The following table describes the type changes and symbols used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>system% You have mail.</code>
AaBbCc123	What you type, contrasted with on-screen computer output	<code>system% su</code> <code>Password: :</code>
<i>AaBbCc123</i>	Command-line placeholder or variable name. Replace with a real name or value	To delete a file, type <code>rm filename</code> . The <i>errno</i> variable is set.
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be root to do this.
Code samples are in code font and may display the following:		
%	UNIX C shell prompt	<code>system%</code>
\$	UNIX Bourne and Korn shell prompt	<code>system\$</code>
#	Superuser prompt, all shells	<code>system#</code>

Introduction to Administration

This chapter introduces you to system administration in the Trusted Solaris environment. It begins with a quick review of Trusted Solaris concepts from the *Trusted Solaris User's Guide* and goes on to explain some advanced concepts necessary for Trusted Solaris administrators.

- “Basic Concepts Review” on page 15
- “Understanding Labels” on page 31
- “Understanding Rights Profiles” on page 21
- “Understanding Roles” on page 20
- “Understanding Authorizations” on page 25
- “Understanding Privileges” on page 29
- “How the Trusted Solaris Environment Controls Device Access” on page 39

Basic Concepts Review

The Trusted Solaris environment is an enhanced version of Solaris that incorporates configurable security policy into the system. The concepts in this section are basic to understanding the Trusted Solaris environment, both for users and administrators. They are briefly covered here and are discussed in more depth in the *Trusted Solaris User's Guide*.

How the Trusted Solaris Environment Protects Against Intruders

Trusted Solaris protects access to the system by providing accounts requiring user names with passwords. Passwords can be created by users or system-generated, according to your site's security policy. You can also require that passwords be changed regularly. In addition, users can work within their approved label range only limiting the information they can access. Additional passwords are required for certain administrative tasks; this limits the damage that can be done by an intruder who guesses the root password.

The Trusted Solaris environment displays the Trusted Path symbol, an unmistakable, tamper-proof emblem that appears at the bottom of the screen, indicating to users when they are using security-related parts of the system. If the Trusted Path symbol does not appear when the user is running a trusted application, that version of the application should be checked immediately for authenticity.

As administrator, you should always verify personally with your users instructions you send them via email. The purpose of this policy is to avoid such situations as imposters posing as administrators and sending email to users to try to get passwords to accounts or other sensitive information.

How the Trusted Solaris Environment Enforces Access Control Policy

The Trusted Solaris environment protects information and other resources through *discretionary access control*—the traditional UNIX permission bits and access control lists set at the discretion of the owner—and *mandatory access control*—a mechanism enforced by the system automatically that controls all transactions by checking the labels of processes and data in the transaction.

A user's *label* represents the sensitivity level at which the user is permitted to and chooses to operate. It determines which information the user is allowed to access. Both mandatory and discretionary access controls can be overridden by special permissions called *privileges*, which are granted to processes. In some cases, users may need *authorizations* as well, which are granted to users (and roles) by an administrator.

As administrator, you need to train users on the proper procedures for securing their files and directories, according to your site's security policy. Furthermore, you should instruct any users allowed to upgrade or downgrade labels as to when it is appropriate to change a label.

How the Trusted Solaris Environment Implements Administration

In conventional UNIX systems, superuser (root) is all-powerful with the ability to read and write to any file, run all programs, and send kill signals to any process. In the Trusted Solaris environment, root's capabilities are divided into separate *role* accounts that can be assigned to different individuals.

Roles are used mainly for security-related tasks. They require separate authentication, are assigned to sysadmin group 14, are privileged NIS+ principals, and operate in special workspaces that can supply the *trusted path attribute* to those processes requiring them; many administrative applications require all four conditions to run successfully.

Understanding Trusted Software Administration

The following sections describe trusted software administration:

- “Overview of Trusted Software Administration” on page 17
- “Understanding Roles” on page 20
- “Understanding Rights Profiles” on page 21
- “Understanding Authorizations” on page 25
- “Understanding Privileges” on page 29

Overview of Trusted Software Administration

As mentioned earlier in this chapter, the superuser's capabilities are divided into separate role accounts rather than being concentrated in the superuser account only. This separation is accomplished through two override mechanisms in the Trusted Solaris environment: *authorizations*, which are rights associated with users, and *privileges*, which are rights associated with processes. An application that can override system controls is called a *trusted application*. Trusted applications have either been designed to check for authorizations or have been assigned special security attributes (that is, effective and real user IDs (UIDs) and group IDs (GIDs), process labels, clearances, and privileges). Instead of becoming superuser, users or roles can run their assigned trusted applications with the same capabilities that the superuser would have running these applications.

Trusted applications and authorizations are grouped in *rights profiles* or *profiles*, for short, which can be assigned to users or more commonly to roles. Users access

trusted CDE actions granted to them through the Front Panel, the Application Manager, the Workspace menu, and the File Manager. Users access trusted commands granted to them through special shells called *profile shells*. The profile shell is a Bourne, Korn, or C shell that has been modified to grant roles (and users) access to those programs assigned to their rights profiles and to make security attributes available to commands. From a profile shell, a user can execute those commands and only those commands assigned to that user's profiles. Note that a profile can be used to *enable* users, that is, give them access to commands, privileges, and authorizations not available to normal users; or to *restrict* users, that is, to limit them to a specific set of commands (this might be appropriate for unsophisticated users).

In practice, here is how access control is enforced and can be overridden by an administrator or authorized user. When a user attempts to access a file, the mandatory access controls (MAC) are checked. The process label of the program the user is running must dominate the label on the file; if this is not the case, then the user's process needs MAC privileges, such as `file_mac_search` to access the directory and `file_mac_read` to read the file. There are also discretionary access controls, that is, UNIX permissions, to be checked. If the user does not have read permission, then privileges such as `file_dac_search` and `file_dac_read` are needed. If the file's ownership were to be changed through the File Manager, then the user would need the authorization `solaris.file.chown`.

The following figure summarizes the elements used in Trusted Solaris administration. The arrows in the figure indicate the direction of an assignment. In short, roles are special accounts that can be assigned to users. Rights profiles are packages of permitted operations that are assigned commonly to roles and occasionally to users. Authorizations, CDE actions, and commands are assigned to rights profiles. Privileges are assigned to sets. The allowed and forced privilege sets can be assigned to executable files. CDE action and command processes can have the following security attributes applied: privileges in the inheritable privilege set; effective and real UIDs/GIDs; and clearances and security labels.

These elements and their relations are discussed in the following sections.

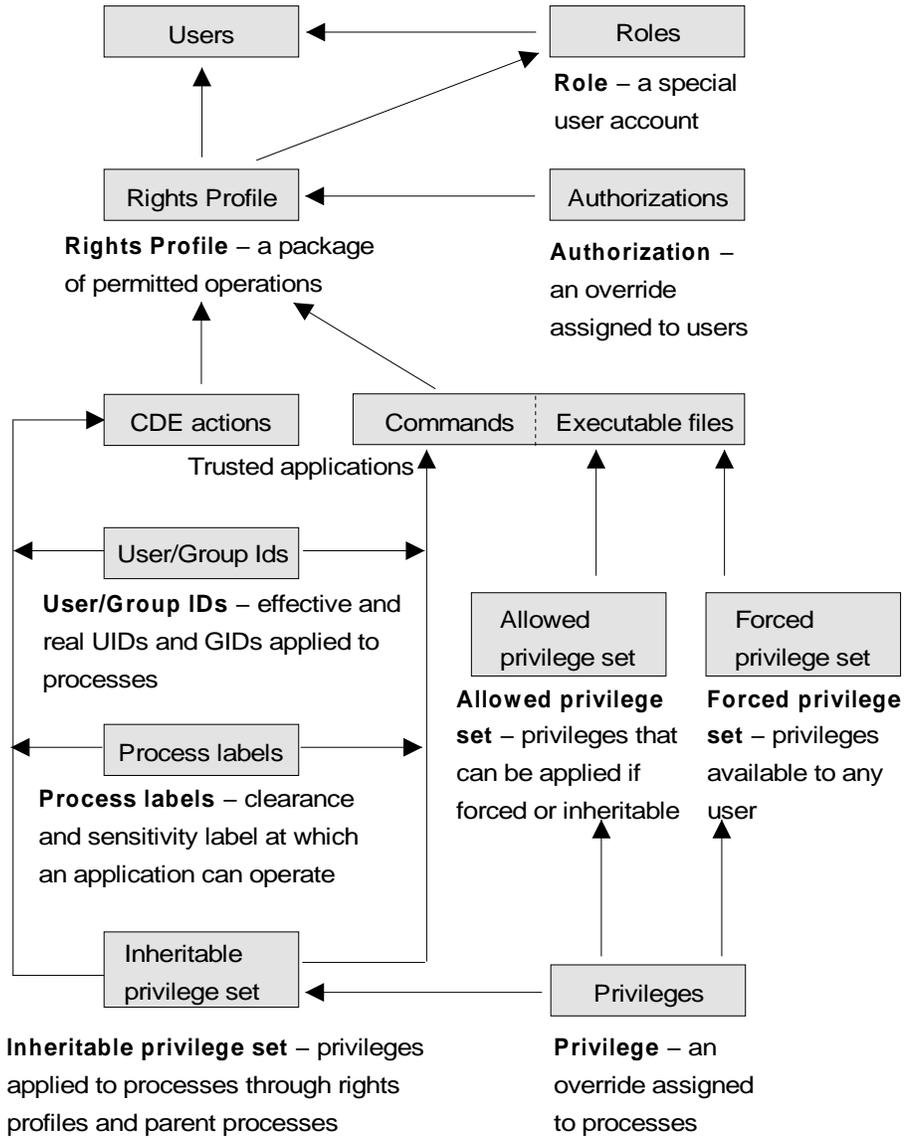


Figure 1-1 Security Element Assignment in the Trusted Solaris Environment

Note - The Solaris environment also provides roles, authorizations, rights profiles, and profile shells (also referred to as “administrator shells”). The only security attributes that trusted applications in the Solaris environment can make use of are real and effective UIDs/GIDs. There are no labels, clearances, or privileges in the Solaris environment.

Understanding Roles

A *role* is a special user account that gives a user access to specific programs and the authorizations and privileges necessary for running them. 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 cannot log in directly to a role; they must log into their user account prior to assuming a role to ensure that the user's real UID is recorded for auditing. (Another restriction that supports auditing is that a user cannot assume any other role directly from a role.) Assuming a role requires users to authenticate themselves by providing the role password. The user is then granted access to a dedicated role workspace where the user has access to trusted applications, the profile shell, and the trusted path attribute.

The Trusted Solaris environment provides one preconfigured role (Root) and four recommended roles as shown in the table below. If your site plans to use these roles, you need to configure them according to the instructions in the "How to Create Administrative Roles" in *Trusted Solaris Installation and Configuration*.

TABLE 1-1 Roles and their Responsibilities

Role (Login Name)	Responsibilities
Root (root)	Initial installation and configuration of the operating environment. After configuration, the root role is not used for administration and should not be assigned to any user.
System administrator (admin)	Performs standard UNIX system administration tasks. Adds new users; configures user templates; modifies certain user properties; configures hosts, networks, routes, and printers. Can also make and restore backups and administer printing.
Security administrator (secadmin)	Responsible for security tasks and decisions. Administers labels; modifies security-relevant attributes of users, networks, printers and other devices and hosts. Configures host templates. Can modify default roles and profiles and add new roles, but cannot grant capabilities beyond those of the security administrator role itself.
Primary administrator (primaryadmin)	Not used in normal system operations, the primary administrator role is designed to be used only when the security administrator role cannot accomplish a task, for example, adding a new role or profile with capabilities the security administrator does not have.
System operator (oper)	Makes backups and administers printing.

The Trusted Solaris environment is highly configurable so that you can implement a customized set of roles and rights profiles. (Note that this type of customization

would be done by the primary administrator role.) If your site does reconfigure roles, make sure all users know who is performing each set of duties.

Your site may need new roles in addition to the predefined administrative roles. The main reason for creating a role is to define an explicit job responsibility that can use special commands and actions and any necessary privileges, that needs to be isolated from normal users, and that uses a shared home directory, files, and environment. (If you need to isolate commands and privileges with separate home directories and files for different users, then you should create a special rights profile instead of a role. See next section.)

Understanding Rights Profiles

Trusted applications and authorizations can be grouped into packages called *rights profiles* for assignment to user or role accounts. The main purpose of a rights profile is to provide limited override power to a user or role who needs this capability.

The potential contents of a rights profile are:

- Authorizations
- CDE actions with or without real and effective UIDs and GIDs, privileges, process labels, and clearances
- Commands with or without real and effective UIDs and GIDs, privileges, process labels, and clearances

Assigning Rights Profiles to Users or Roles

To assign rights profiles to users, you open the User Properties dialog box from the User Tool in the Solaris Management Console and select the Rights tab, as shown in the following figure. The rights profiles not assigned to the current user or role are displayed in the Excluded column on the left and must be moved to the right column for assignment to the current account. For more information, see the online help. In similar fashion, you can make changes to roles through the Administrative Roles dialog box in the User Tool.

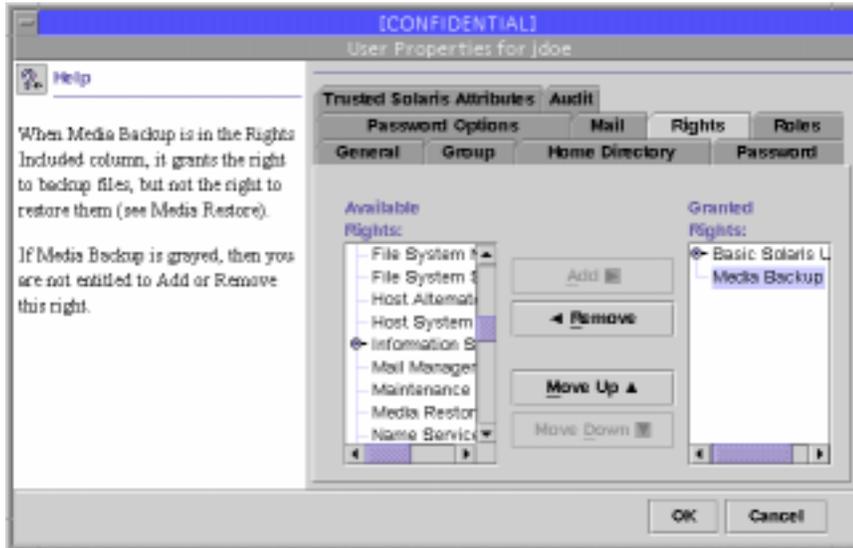


Figure 1-2 Assigning Rights Profiles to Users

Predefined Trusted Solaris Rights Profiles

The Trusted Solaris environment provides a set of predefined rights profiles (see the following table). Before you assign any of these rights profiles, you should familiarize yourself with their contents. To view the contents of predefined rights profiles, use the `-list` option in the `smprofile` command (see next section) or the Rights dialog box. The profiles can be modified according to the needs of your organization.

TABLE 1-2 Rights Profile Descriptions

Rights Profile	Purpose
All	Provides access to all executables but without privileges.
All Actions	Provides access to all actions but without privileges.
All Authorizations	Provides all authorizations. For testing.
All Commands	Provides access to all commands but without privileges.
Audit Control	For managing the audit subsystem but without ability to read files.
Audit Review	For reading the audit trail.
Basic Actions	Provides access to the applications on the Front Panel with the necessary privileges.

TABLE 1-2 Rights Profile Descriptions *(continued)*

Rights Profile	Purpose
Basic Commands	Provides access to rudimentary commands necessary for all roles.
Basic Solaris User	Assigned to all users of the Solaris Management Console. Provides Read permissions and lets users add con jobs to their crontab files. Contains All rights profile.
Convenient Authorizations	Provides authorizations for normal users.
Cron Management	For managing cron and at jobs.
Custom Admin Role	This is an empty right for adding security attributes to the default Admin role.
Custom Oper Role	This is an empty right for adding security attributes to the default Oper role.
Custom Root Role	This is an empty right for adding security attributes to the default Root role.
Custom Secadmin Role	This is an empty right for adding security attributes to the default Secadmin role.
Custom SSP	This is an empty right for adding security attributes to the default SSP role for Sun Enterprise 10000 administration.
Device Management	For allocating and deallocating devices, and correcting error conditions.
Device Security	For managing and configuring devices.
Enable Login	Provides the authorization for allowing yourself and other users to log in after boot.
File System Management	For managing file systems.
File System Security	For managing file system labels and other security attributes.
Information Security	For setting access control policy.
Mail Management	For configuring sendmail, modifying aliases, and checking mail queues.
Maintenance and Repair	Provides commands needed to maintain or repair a system.
Media Backup	Backup files.
Media Restore	Restore files from backup.
Name Service Management	Grants right to control the name service daemon.
Name Service Security	Grants right to control the name service properties and table data.

TABLE 1-2 Rights Profile Descriptions *(continued)*

Rights Profile	Purpose
Network Management	For managing the host and network configuration.
Network Security	For managing network and host security, with authorizations for modifying trusted network databases.
Object Access Management	For changing ownership and permissions on files.
Object Label Management	For changing labels of files and setting up system-wide labels.
Object Privilege Management	For changing privileges on executable files.
Outside Accredited	Operate outside system accreditation range.
Primary Administrator	Contains subordinate rights profiles for primary administrator role.
Privileged Shells	For developers to run Bourne, Korn, and C shells with all privileges. NOT intended for secure environments.
Process Management	For managing current processes, including cron and at jobs.
Remote Administration	Remote administration of headless systems.
Rights Delegation	Lets user or role assign rights assigned to that user or role to other users or roles. Lets user assign roles assigned to that user to other users.
Rights Security	For managing assignment of rights profiles, labels, and privileges, and for setting account security.
Software Installation	For adding application software to the system.
SSP Administration	Tools for administering the SSP.
SSP Installation	Tools for installing the SSP.
System Administrator	Contains subordinate rights profiles for system administrator role.
User Management	For creating and modifying users but without the ability to modify self (as a security measure).
User Security	For creating and modifying users' security attributes but without the ability to modify self (as a security measure).

Displaying Rights Profile Information

Use the `-list` option in the `smprofile` command to obtain various rights profile information. This command lets you display the contents of any profile for all users or specified user(s) and optionally the contents of the profiles. Another option for displaying rights profile information is `profiles(1)`.

Customizing Rights Profiles

If the predefined rights profiles as they are shipped are not appropriate for your organization, they can be modified by the security administrator (or other role with equivalent powers). The Rights dialog box is used to edit the contents of rights profiles (see figure below). The Rights dialog box is accessed from the User Tool in the Solaris Management Console. For more information, see the online help.

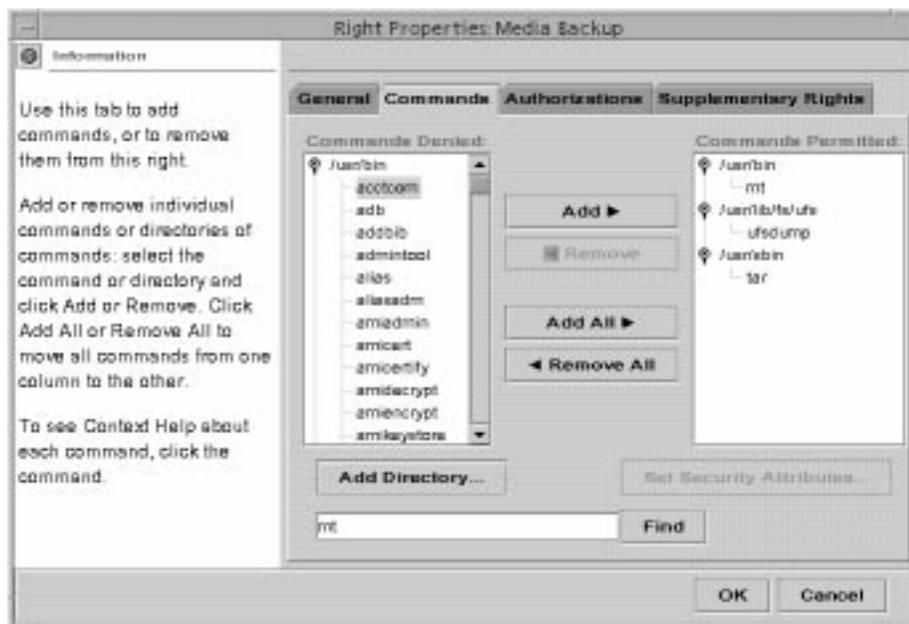


Figure 1-3 Rights Dialog Box

Understanding Authorizations

An *authorization* is a discrete right granted to a user or role that is checked by certain trusted applications to determine whether the user is permitted to execute a restricted function. For example, in a conventional system, the file manager allows superuser only to change the ownership of a file. In the Trusted Solaris operating environment, the authorization `Change File Owner` is required.

An authorization has a name, which is used internally and in files (for example, `solaris.file.owner`), and a short description, which appears in the graphical interfaces (for example, `Act as File Owner`). By convention, authorization names begin with the reverse order of the internet name followed by the subject area, any subarea, and the function, all separated by dots, for example, `com.xyzcorp.device.access`. The exceptions to this convention are authorizations from Sun Microsystems, Inc., 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.

The authorizations provided in the Trusted Solaris environment are shown in the following table.

TABLE 1-3 Authorizations

Authorization Category	Authorization Name — Short Description
solaris.admin.dcmgr.*	solaris.admin.dcmgr.admin—Manage OS Services and Patches
	solaris.admin.dcmgr.clients—Manage Diskless Clients
	solaris.admin.dcmgr.read—View OS Services, Patches and Diskless Clients
solaris.admin.diskmgr.*	solaris.admin.diskmgr.read—View Disks
	solaris.admin.diskmgr.write—Manage disks
solaris.admin.fsmgr.*	solaris.admin.fsmgr.write—Mount and Share Files
	solaris.admin.fsmgr.read—View Mounts and Shares
solaris.admin.logsvc.*	solaris.admin.logsvc.write—Manage Log Settings
	solaris.admin.logsvc.purge—Remove Log Files
	solaris.admin.logsvc.read - View Log Files
solaris.admin.nameservice.*	solaris.admin.nameservice.config—Name Service Configuration
solaris.admin.printer.*	solaris.admin.printer.read—View Printer Information
	solaris.admin.printer.modify—Update Printer Information
	solaris.admin.printer.delete—Delete Printer Information
solaris.admin.procmgr.*	solaris.admin.procmgr.admin—Manage All Processes
	solaris.admin.procmgr.user—Manage Owned Processes

TABLE 1-3 Authorizations (continued)

Authorization Category	Authorization Name — Short Description
solaris.admin.serialmgr.*	solaris.admin.serialmgr.modify—Manage Serial Ports solaris.admin.serialmgr.delete—Delete Serial Ports solaris.admin.serialmgr.read—View Serial Ports
solaris.admin.usermgr.*	solaris.admin.usermgr.audit—Set User Audit Info solaris.admin.usermgr.write—Manage Users solaris.admin.usermgr.psword—Change Password solaris.admin.usermgr.read—View Users and Roles solaris.admin.usermgr.labels—Set User Label Info
solaris.audit.*	solaris.audit.config—Configure Auditing solaris.audit.read—Read Audit Trail
solaris.compsys.*	solaris.compsys.read—View Computer System Information solaris.compsys.write—Manage Computer System Information
solaris.device.*	solaris.device.allocate—Allocate Device solaris.device.config—Configure Device Attributes solaris.device.grant—Delegate Device Administration solaris.device.revoke—Revoke or Reclaim Device
solaris.file.*	solaris.file.audit—Set File Audit Attributes solaris.file.chown—Change File Owner solaris.file.privs—Set File Privilege solaris.file.owner—Act as File Owner
solaris.grant	solaris.grant—Grant All Solaris Authorizations
solaris.jobs.*	solaris.jobs.admin—Manage All Jobs solaris.jobs.grant—Delegate Cron & At Administration solaris.jobs.user—Manage Owned Jobs

TABLE 1-3 Authorizations (continued)

Authorization Category	Authorization Name — Short Description
solaris.label.*	solaris.label.print—View Printer Queue at All Labels
	solaris.label.file.downgrade—Downgrade File Label
	solaris.label.file.upgrade—Upgrade File Label
	solaris.label.range—Set Label Outside User Accred Range
	solaris.label.win.downgrade—Downgrade DragNDrop or CutPaste Info
	solaris.label.win.noview—DragNDrop or CutPaste without viewing contents
	solaris.label.win.upgrade—Upgrade DragNDrop or CutPaste Info
solaris.login.*	solaris.login.enable—Enable Logins
	solaris.login.remote—Remote Login
	solaris.login.su—Switch User Without Trusted Path
solaris.network.*	solaris.network.hosts.read—View Computers and Networks
	solaris.network.hosts.write—Manage Computers and Networks
	solaris.network.security.write—Manage Trusted Networking
	solaris.network.security.read—View Trusted Networking
solaris.print.*	solaris.print.admin—Administer Printer
	solaris.print.list—List Jobs in Printer Queue
	solaris.print.cancel—Cancel Print Job
	solaris.print.nobanner—Print without Banner
	solaris.print.ps—Print Postscript
	solaris.print.unlabeled—Print without Label
solaris.profmgr.*	solaris.profmgr.assign—Assign All Rights
	solaris.profmgr.delegate—Assign Owned Rights
	solaris.profmgr.execattr.write—Manage Commands
	solaris.profmgr.read—View Rights
	solaris.profmgr.write—Manage Rights

TABLE 1-3 Authorizations (continued)

Authorization Category	Authorization Name — Short Description
solaris.role.*	solaris.role.assign—Assign All Roles
	solaris.role.delegate—Assign Owned Roles
	solaris.role.write—Manage Roles
solaris.system.*	solaris.system.date—Set Date & Time
	solaris.system.shutdown—Shutdown the System

For a complete list of authorizations, see the `/etc/security/auth_attr` file. Authorizations are assigned to rights profiles using the Rights dialog box in the SMC User Manager.

Understanding Privileges

A *privilege* is a discrete right granted to a process to perform an operation that would otherwise be prohibited by the Trusted Solaris environment. For example, 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.

How a Process Acquires Privileges

The Trusted Solaris environment determines which privileges a process can make effective based on the allowed and forced privilege sets assigned to the executable file and the inheritable privileges inherited by the process.

The *allowed privilege* attribute satisfies one condition necessary for that privilege to be effective. If an allowed privilege for an application is not set, the privilege cannot be effective under any condition. The *forced privilege* attribute makes the privilege effective to all users running that application. Both types of attributes are assigned using either the File Manager or the `setfpriv(1)` command. The command `getfpriv(1)` lets you see which privileges are set on the executable file. Note that if an executable file is modified, all allowed and forced privileges are removed.

The *inheritable privilege* attribute is assigned to the application within a rights profile. Only users who have been assigned that rights profile are granted the privilege for that application. Inheritable privilege attributes are assigned to an application inside a rights profile using either the Rights Manager or the `-add` option in the `smexec` command. An inheritable privilege is made effective when the process is launched by one of the trusted launchers. For the terminal environment, the Trusted Solaris

environment provides three profile shells corresponding to the Bourne, Korn and C shells; for the desktop, the Workspace Menu, the Front Panel, and the Application Manager interpret profiles for actions; and for remote environments the Solaris Management Console legacy application tool interprets profiles. A process can also pass inheritable privileges to any program it executes, provided that the particular privilege is *allowed* by the program.

Note - In contrast to inheritable privileges, forced privileges cannot be inherited by child processes except in applications that have been customized especially for the Trusted Solaris environment to have that specific capability. To provide privileges to a shell script, one should thus use inheritable privileges, not forced privileges.

Default Privileges Supplied by the Trusted Solaris Environment

The Trusted Solaris environment provides more than 80 privileges that you can apply to applications to override security policy. For a complete list of privileges, see the `priv_desc(4)` man page. The privileges provided fall into the categories shown in the following table.

TABLE 1-4 Privilege Categories

Privilege Category	Summary	Example Privileges in the Category
File system security	For overriding file system restrictions on user and group IDs, access permissions, labeling, ownership, and file privilege sets	<i>file_dac_chown</i> – lets a process change the owner user ID of a file.
System V Interprocess Communication (IPC) security	For overriding restrictions on message queues, semaphore sets, or shared memory regions	<i>ipc_dac_read</i> – lets a process read a System V IPC message queue, semaphore set, or shared memory region whose permission bits or ACL do not allow process read permission
Network security	For overriding restrictions on reserved port binding or binding to a multilevel port, sending broadcast messages, or specifying security attributes (such as labels, privileges on a message, or network endpoint defaults)	<i>net_broadcast</i> – lets a process send a broadcast packet on a specified network
Process security	For overriding restrictions on auditing, labeling, covert channel delays, ownership, clearance, user IDs, or group IDs	<i>proc_mac_read</i> – lets a process read another process where the reading process label is dominated by the other process label

TABLE 1-4 Privilege Categories (continued)

Privilege Category	Summary	Example Privileges in the Category
System security	For overriding restrictions on auditing, workstation booting, workstation configuration management, console output redirection, device management, file systems, creating hard links to directories, increasing message queue size, increasing the number of processes, workstation network configuration, third-party loadable modules, or label translation	<i>sys_boot</i> – lets a process halt or reboot a Trusted Solaris workstation
Window security	For overriding restrictions on colormaps, reading to and writing from windows, input devices, labeling, font paths, moving data between windows, X server resource management, or direct graphics access (DGA) X protocol extensions	<i>win_selection</i> – allows a process to request inter-window data moves without the intervention of selection arbitrator

Understanding Labels

Labels and clearances are the heart of mandatory access control in the Trusted Solaris environment. They determine which users can access which files and directories.

Labels and clearances consist of one *classification* component and zero or more *compartment* components. The classification component indicates a hierarchical level of security such as TOP SECRET or CONFIDENTIAL. The compartment component represents a group of users who may need access to a common body of information. Some typical types of compartments are projects, departments, or physical locations.

The Trusted Solaris environment mediates all attempted security-related transactions. It compares the labels of the accessing entity, typically a process, and the entity being accessed, usually a file, and then permits or disallows the transaction depending on which label is *dominant* (as described in the following section). Labels are also used to determine access to other system resources, such as allocatable devices, networks, framebuffer, and other hosts.

Note - CMW labels are primarily of importance to programmers. They are composed of regular labels (also called sensitivity labels) and an obsolete label type called an information label. Although they are present in CMW labels (for backwards compatibility), information labels are no longer used by the system.

Dominance Relationships Between Labels

One entity's label is said to *dominate* another's if the following two conditions are met:

- The classification component of the first entity's label is equal to or higher than the second entity's classification. (The security administrator assigns numbers to classifications in the `label_encodings(4)` file; these numbers are compared when determining dominance.)
- The set of compartments in the first entity includes all of the second entity's compartments.

Two labels are said to be *equal* if they have the same classification and the same set of compartments. If they are equal, they dominate each other and access is permitted.

If one label has a higher classification or if it has the same classification and its compartments are a superset of the second label's compartments or both, the first label is said to *strictly dominate* the second label.

Two labels are said to be *disjoint* or *noncomparable* if neither label dominates the other.

The following table presents examples of label comparisons for dominance. In the example, `NEED_TO_KNOW` is a higher classification than `INTERNAL`. There are three compartments: Eng, Mkt, and Fin.

TABLE 1-5 Examples of Label Relationships

Label 1	Relationship	Label 2
NEED_TO_KNOW Eng Mkt	(strictly) dominates	INTERNAL Eng Mkt
NEED_TO_KNOW Eng Mkt	(strictly) dominates	NEED_TO_KNOW Eng
NEED_TO_KNOW Eng Mkt	(strictly) dominates	INTERNAL Eng
NEED_TO_KNOW Eng Mkt	dominates (equals)	NEED_TO_KNOW Eng Mkt
NEED_TO_KNOW Eng Mk	is disjoint with	NEED_TO_KNOW Eng Fin
NEED_TO_KNOW Eng Mkt	is disjoint with	NEED_TO_KNOW Fin
NEED_TO_KNOW Eng Mkt	is disjoint with	INTERNAL Eng Mkt Fin

Administrative Labels

The Trusted Solaris environment provides two special labels for administration to be used as labels or clearances: `ADMIN_HIGH` and `ADMIN_LOW`. (You can rename

these two labels in the `label_encodings(4)` file if you choose.) These labels are used to protect system resources and are intended for administrators rather than normal users.

ADMIN_HIGH is the highest label; it dominates all other labels in the system and is used to protect system data, such as administration databases or audit trails, from being read. You need to work at the ADMIN_HIGH label (typically in a role) or have the privilege to read up from your current label to read data labeled ADMIN_HIGH.

ADMIN_LOW is the lowest label; it is dominated by all other labels in a system.

Mandatory access control does not permit users to write data to files with labels lower than the subject's label. Thus, applying ADMIN_LOW, the lowest label, to a file ensures that normal users cannot write to it although they can read it.

ADMIN_LOW is typically used to protect public executables and configuration files to prevent them from being modified, since only a user working at ADMIN_LOW or with the privilege to write down would be able to write to these files. Typically, only an administrator would work at ADMIN_LOW.

Label Encodings Files

All label components for a system, that is, classifications, compartments, and the associated rules are stored in a file called `label_encodings(4)` (located in `/etc/security/tsol`). The security administrator sets up the `label_encodings` file for the site. A label encodings file contains:

- component definitions—definitions of classifications, compartments, labels, and clearances, including rules for required combinations and constraints
- accreditation range definitions—specification of the clearances and minimum labels that define the sets of available labels for the entire system and for normal (non-administrative) users
- printing specifications—identification and handling information for print banners, trailers, headings, footers, and other security features for printouts
- customizations—local definitions including label color codes, alternative names for classifications, compartments, and markings in the graphical interface, and other items

For more information on the `label_encodings` file, see the man page for `label_encodings(4)` and the manuals, *Trusted Solaris Label Administration* and *Compartmented Mode Workstation Labeling: Encodings Format*.

Label Ranges

A *label range* is the set of potentially usable labels at which users can operate. Resources that can be protected by label ranges include such things as allocatable devices, file systems, networks, interfaces, frame buffers (effectively workstations),

and commands or actions. A label range is defined by a clearance at the top of the range and a minimum label at the bottom. A range is not necessarily all combinations of labels that fall between a maximum and minimum label. There may be rules in the label encodings file that disqualify certain combinations. A label must be *well-formed*, that is, permitted by all applicable rules in the label encodings file, in order to be included in a range. On the other hand, a clearance does not have to be well-formed. Suppose, for example, that a label encodings file prohibits any combination of compartments Eng, Mkt, and Fin in a label. INTERNAL Eng Mkt Fin would be a valid clearance but not a valid label; as a clearance, it would let a user access files labeled INTERNAL Eng, INTERNAL Mkt, and INTERNAL Fin.

Account Label Range

When you assign a clearance and a minimum label to a user, you define the upper and lower boundaries of the *account label range* in which that user is permitted to operate. The following equation describes the account label range, using \leq to indicate dominated by or the same as:

minimum label \leq permitted label \leq clearance

Thus, the user is permitted to operate at any label that is dominated by the clearance as long as that label is not strictly dominated by the minimum label. If you do not expressly set a user's clearance or minimum label, the defaults defined in the `label_encodings` file will take effect. Make sure when you assign a clearance that the classification dominates (or is the same as) all classifications at which the user can work and that the list of compartments include all compartments that user might need. Combinations of compartments in the clearance will be governed by rules in the `label_encodings` file.

To assign single-label operation to a user, you set the user's clearance equal to the minimum label.

Session Range

The *session range* is the set of labels available to a user during a Trusted Solaris session. The session range must be within the user's account label range and the label range set for the system. If the user selects single-label session mode, the session range will be limited to that label. If the user selects multilabel mode, then the label entered will serve as the session clearance, defining the upper boundary of the session range while the user's minimum label defines the lower bound. The user enters the session at the minimum label and can switch to a workspace at any label in the session range.

How Labeled Files are Stored

In the Trusted Solaris environment, labels are automatically associated with all files and directories, and are stored as extended attributes of the file. These attributes are protected by privilege and mandatory controls.

In addition, special directories called *multilevel directories (MLDs)* allow files to be isolated by label in subdirectories called *single-level directories (SLDs)*. SLDs are transparent to users and applications.

The purpose of MLDs is to enable applications that are running at different labels to write into what appears to be the same directory. For example, the `/tmp` directory is often used by multiple applications; for that reason, `/tmp` is an MLD. Applications are not aware that when they write a file into `/tmp` they are actually writing the file into the SLD within `/tmp` that has the label at which the application is running. If a single-level directory corresponding to the label does not yet exist, the Trusted Solaris environment creates one automatically.

New MLDs are built by creating a new folder with the File Manager using the MLD option or at the command line using the `--M` option of the `mkdir(1)`. The `crontab(1)` and at-job directories are shipped as MLDs so that you can set up batch jobs for a user that run at different labels. See the “Administering the Automatic Running of Jobs Using cron, at, and batch” in *Trusted Solaris Administrator’s Procedures*.

Home directories are MLDs so that accounts can create files and folders at different labels within their home directories. When user or role accounts change into their home directories, they do not need to be aware that they have actually changed into an SLD that is at the same label as their current workspace. For example, when setting up a new account for user `roseanne`, the User Tool creates the home directory `/export/home/roseanne` as an MLD. When the user `roseanne` changes to her home directory, she is automatically and transparently redirected to an SLD within her home directory MLD. The SLD has the same label as her current workspace, so if the workspace has a label of `NEED_TO_KNOW`, she changes into the SLD that has the `NEED_TO_KNOW` label.

To allow normal users to create their own MLDs, the administrator role must first create a new directory that is not an MLD and make it writable by normal users. For example, an administrator could create a directory called `/myDir/doc` mounted by and writable by all developers at a single label, so that design specifications and other project-wide documentation could be kept in one commonly accessible place. Anyone in the development group could then create a new directory within that directory and make it an MLD. If desired, the prefix can be changed from MLD using the `mount(1M)` command.

Multilevel directory names contain a hidden string, `.MLD.` (referred to as an *adornment*), which is appended to the beginning of the directory name but is not visible to standard UNIX commands.

Single-level directories are named `.SLD.n` where the number `n` represents the order in which the SLDs in the multilevel directory are created. Thus, the single-level

directories are named `.SLD.0`, `.SLD.1`, and so on. The implementation is transparent so that directory names with adornments are not displayed except through the special commands in the table below. A user with appropriate privileges can view the contents of a hidden directory outside of the current SL by explicitly specifying the adornments to the path.

TABLE 1-6 Adornment—Related Commands

Command Name	Description
<code>adornfc(1)</code>	The <code>adornfc(1)</code> command displays the specified directory pathname with the final component adorned, that is, the strings <code>.MLD.</code> or <code>.SLD.</code> used to identify whether the directory is multilevel or single-level.
<code>getfattrflag(1)</code>	The <code>-m</code> option indicates whether or not the directory is an MLD.
<code>getmldadorn(1)</code>	The <code>getmldadorn(1)</code> command displays the MLD adornment of the filesystem on which the specified pathname resides.
<code>getslname(1)</code>	The <code>getslname(1)</code> command displays the single-level directory name associated with the label of the current process within the multilevel directory referred to by pathname.
<code>mkdir(1)</code>	When used with <code>-M</code> option or when the directory name has the <code>.MLD.</code> adornment, creates a new MLD.
<code>mldpwd(1)</code>	The <code>mldpwd(1)</code> command displays the pathname of the current working directory, including any MLD adornments and SLD names.
<code>mldrealpath(1)</code>	The <code>mldrealpath(1)</code> command displays the canonicalized absolute pathname, including any MLD adornments and SLD names. It expands all symbolic links and resolves references to special characters (<code>/.</code> and <code>/..</code>) and translations in pathnames. The resulting path has no special characters, unadorned multilevel directories, or any hidden SLD names.
<code>rm(1)</code> , <code>rmdir(1)</code>	The <code>-M</code> option when used with the <code>-R</code> option removes SLD subdirectories recursively.

The following figure illustrates the normal view of an SLD, depicting directories as ovals, files as rectangles, visible items with solid lines and bolding, and hidden items

with dashed lines and normal font. In this case, the user is operating with a NEED_TO_KNOW Eng Mkt label and executes the `ls` command as shown on the left side of the figure. The user can view files with the Top Secret label only. The actual structure and contents of `myHomeDir`, which is a multilevel directory, is shown at the right of the figure.

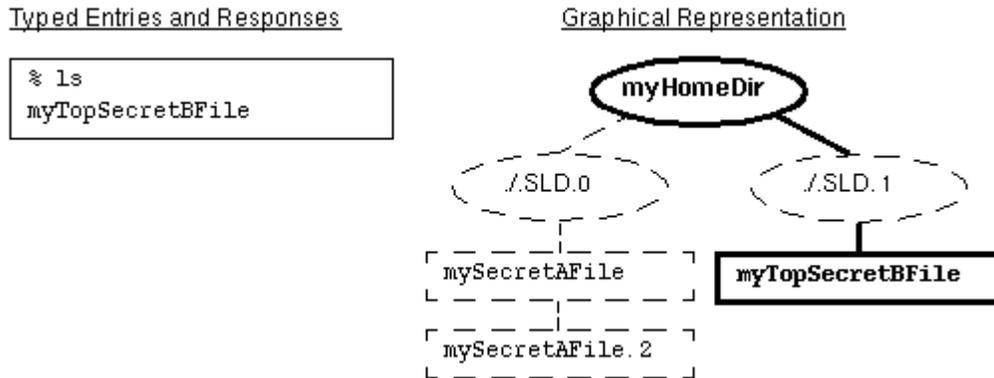


Figure 1-4 Normal Viewing of a Directory

The following figure demonstrates how a user can view directory contents outside of the current SL. By typing `ls /.MLD.myHomeDir/.SLD.*`, the user sees all hidden directories in the multilevel directory, in this case, `.SLD.0` which contains files with an SL of INTERNAL Eng and `.SLD.1` which holds a TOP SECRET file.

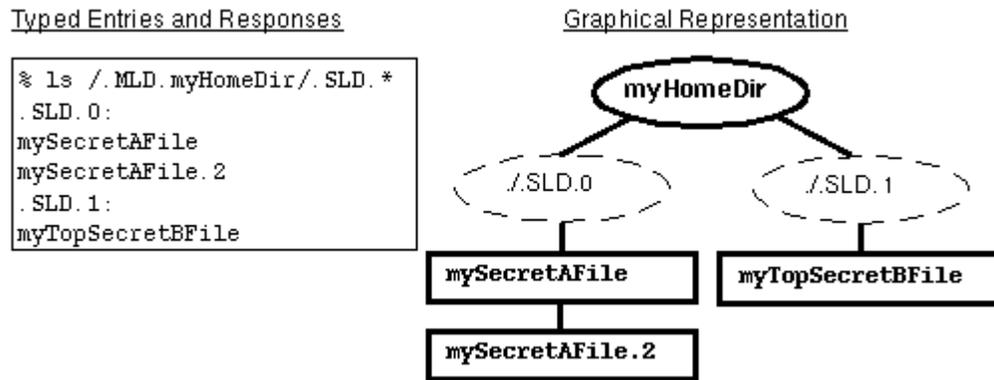


Figure 1-5 Viewing the Contents of Multiple SLDs

Applying Labels to Email

All email messages have labels in the Trusted Solaris environment. The underlying tool `sendmail(1M)` does not deliver mail to a user outside of the user's account

range; use the `-p` option in the `sendmail.cf` file to provide for out-of-range mail. Furthermore, the `restrictmailq` option in the `sendmail.cf` file is set by default to restrict users from listing mail sent by other users; only users in the same group as the mail queue can list jobs in the queue. Email operations make use of multilabel directories both for messages queued prior to delivery and for storage of incoming messages. Users are notified separately about mail received at each label in their account range and in the range of any role they have assumed. In addition, an option exists to promote email from administrators from `ADMIN_LOW` to the user's minimum label.

Applying Labels to Printed Output

You can arrange for labels, handling information, and other security information to be printed out in the banner and trailer pages on a printer by printer basis. The following figure shows a typical banner page. For more information on configuring printing in the Trusted Solaris environment, see “Managing Printing” in *Trusted Solaris Administrator's Procedures* and “Configuring How Labels are Printed on Banner/Trailer and Body Pages” in *Trusted Solaris Label Administration*.

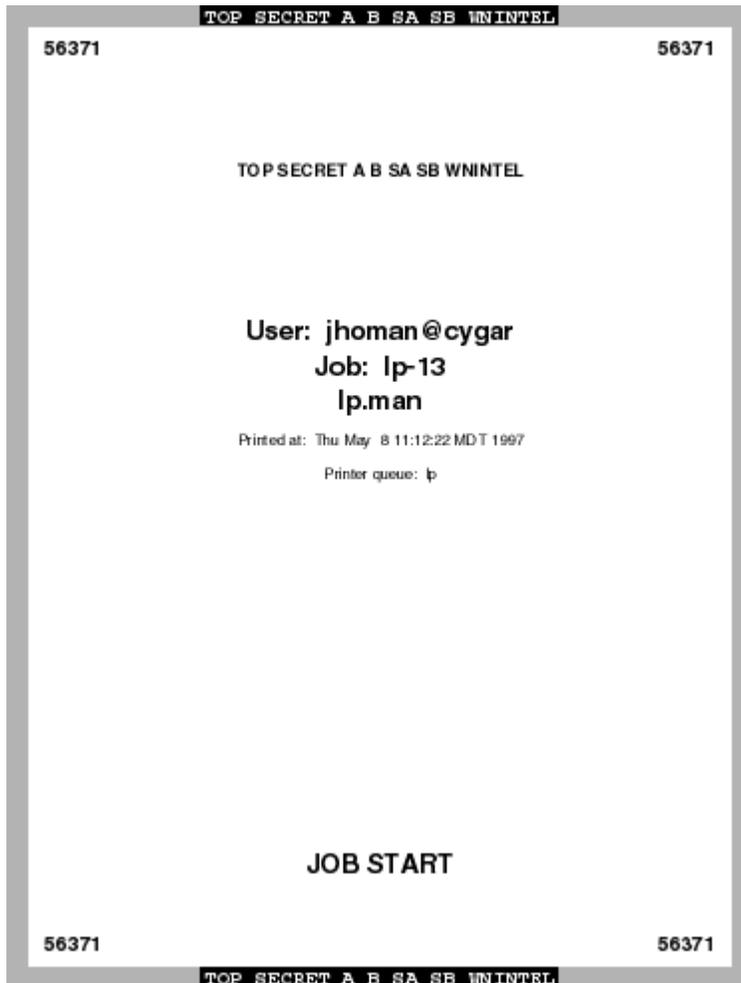


Figure 1-6 Typical Print Banner Page

How the Trusted Solaris Environment Controls Device Access

Since devices provide a means for the import and export of data to and from a Trusted Solaris system, they must be controlled to properly protect the data. (A *device* is either a physical peripheral that is connected to a Trusted Solaris system or a software-simulated device called a pseudo-device.) The Trusted Solaris environment

lets you control data flowing through devices through device allocation and device label ranges.

Device Allocation

Device allocation provides a way to control data when it is imported and exported and prevents unauthorized users from access to the information. In a Trusted Solaris system the administrator decides which devices, if any, each user can use to import and export data and sets those devices to be allocatable. The administrator then assigns to selected users the `Allocate Device` authorization. The `Configure Device Attributes`, `Delegate Device Administration`, and `Revoke or Claim Device` authorizations are used to administrate devices. Users authorized to use a device must allocate the device before using it and deallocate the device when finished. Between the allocation and deallocation of a device, the user has exclusive use of it.

The device allocation applications are provided by the Solaris SunSHIELD Basic Security Module (BSM); refer to Chapter 4, “Device Allocation,” in the *SunSHIELD Basic Security Module Guide*. The Trusted Solaris environment provides a graphical user interface on top of these commands called the Device Allocation Manager that enables device label ranges.

Device allocation provides a way to control the import and export of data. In the Trusted Solaris environment, the administrator decides which devices, if any, can be used to import and export data and includes the devices in the `device_maps(4)` file.

Users allocate devices through the Device Allocation Manager. The Device Allocation Manager mounts the device, runs a clean script to prepare the device and performs the allocation. When finished, the user deallocates the device through the Device Allocation Manager, which runs another clean script and unmounts and deallocates the device.

Device Label Ranges

To prevent users from copying off sensitive information, each allocatable device has an associated label range that is assigned by an administrator. To use an allocatable device, the user must be currently operating at a label within the device’s label range; if not, allocation is denied. The user’s current label is applied to data imported or exported while the device is allocated to the user. The label of exported data is displayed when the device is deallocated so that the user can physically label the medium containing the exported data.

Examples of devices that have label ranges are frame buffers, tape drives, diskette and CD-ROM drives, printers, and network interfaces.

Administering Devices through the Device Allocation Manager

The Device Allocation Manager is accessed from the Tools subpanel above the Style Manager in the Front Panel. The Device Allocation Manager is available to users with the `Allocate Device` authorization for allocation and deallocation only. Normal users cannot see if a device is currently allocated to another user and cannot perform maintenance through the Device Administration button in the Device Allocation Manager, which is available to authorized users and administrators only. The Device Allocation Manager is shown in the following figure.



Figure 1-7 Device Allocation Administration Dialog Boxes

Device Administration Dialog Box

Clicking the Device Administration button in the Device Allocation Manager main window causes the Device Administration dialog box to be displayed (see following figure). The Device Administration dialog box lets you select a device. Its state is then displayed. The buttons in the upper right of the dialog box let you perform operations on the selected device. Clicking the Revoke button moves the selected device from a busy (allocated) state to an available (deallocated) state. Clicking the Reclaim button lets you make available a device that is currently in an error state. The revoke or reclaim device authorization is required to use these buttons. Clicking the Delete button makes a device unavailable. Clicking the New or Configure buttons displays the Device Allocation Configuration dialog box.



Figure 1-8 Device Administration Dialog Box

Device Allocation Configuration Dialog Box

To use the Device Allocation Configuration dialog box requires the `configure device attributes` authorization. Clicking the Configuration button in the Device Allocation Maintenance dialog box causes the Device Allocation Configuration dialog box to be displayed (see following figure).



Figure 1-9 Device Allocation Configuration Dialog Box

The Device Allocation Configuration dialog box is divided into three parts:

- Device security attributes—includes device name and type, minimum and maximum labels, clean program, and device map.
- Allocation specifications—from Trusted Path or non-Trusted Path (for command line users), authorized users (with the authorizations specified in the Authorizations field), no users (if device is not allocatable), all users (if no authorizations required), and which authorizations to require for device allocation
- Deallocation options—deallocate any allocated devices on reboot and deallocate any allocated devices on logout

Device Allocation Authorizations Dialog Box

If you click the Authorizations button in the Device Allocation Configuration dialog box, the Device Allocation Authorizations dialog box is displayed (see following figure). It lets you specify the authorizations required for using the device.

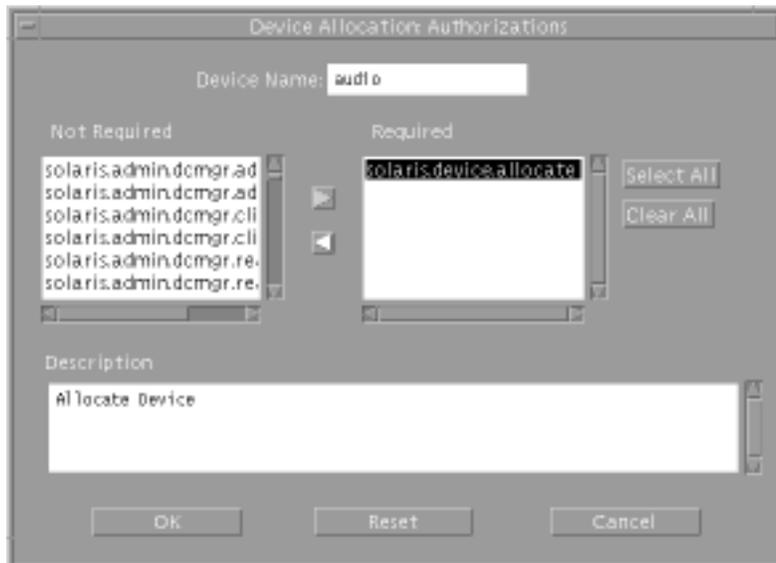


Figure 1–10 Device Allocation Authorizations Dialog Box

Device Allocation Databases and Commands

If you do not have access to the Device Allocation Manager, you can use the commands below to administer allocatable devices. The commands use the device databases: `device_allocate(4)`, `device_deallocate(4)`, and `device_maps(4)`. Note that the commands are not intended for non-administrative users.

- `add_allocatable(1M)`—adds devices to the allocation databases.
- `allocate(1M)`—manages the ownership of devices through its allocation mechanism. It ensures that each device is used by only one qualified user at a time.
- `deallocate(1M)`—deallocates a device allocated to the evoking user.
- `list_devices(1M)`—lists the allocatable devices in the system according to specified qualifications.
- `dminfo(1M)`—displays information about device entries in the device maps file.

Device Clean Scripts

Device clean scripts are special scripts that are run when a device is first allocated. Clean scripts address two security concerns:

- Object reuse – the requirement that a device is clean of previous data before being allocated or reallocated

- Media labeling – the requirement that removable information storage media have a physical label indicating its label. While the ultimate responsibility for putting the labels on the removable media rests with the user, the device clean scripts can prompt the user to do so.

The name of a device clean script for a specific device is stored with that device's entry in the `device_allocate(4)`, file. The operations of each device clean program are specific to each device. The following is a list of tasks that a device clean program performs:

- Eject media – Devices that store information on removable media must be forced to eject that media upon deallocation or reallocation of the device, to prevent passing information to the next user of the device who may be at a different label.
- Reset device state – Devices that keep state information can potentially be used as a covert channel by the users. Thus driver status information must be reset to default values during deallocation of the device.
- Remind user about media labeling – It is a requirement that removable information storage media be labeled with appropriate external media labels. The device user's label is passed to the device clean program when it is invoked (See `device_clean(1M)` man page for interface detail.)

Not all allocatable devices require a device clean program. Devices that do not keep states and do not use removable media do not need a device clean program.

Device clean programs for tape, floppy disk, CD-ROM, and audio devices are provided by the Trusted Solaris environment. The configurable nature of the user device allocation mechanism lets an administrator install new devices and configure device clean programs accordingly.

Device Allocation Security Policy

For more information on device allocation, see Chapter 15, “Managing Devices,” in *Trusted Solaris Administrator's Procedures*.

Quick Tour of the Admin Tools

This chapter presents an overview of the tools available in the Trusted Solaris environment, how they are accessed, and the databases on which they operate.

Introduction to Trusted Solaris Administration

Administration in the Trusted Solaris operating environment uses many of the same tools available in the Solaris operating environment and offers security-enhanced tools as well. The difference between the environments lies in how administration tools are accessed and how this access is restricted.

Accessing Tools in a Role Workspace

To use the Trusted Solaris administration tools, you must be in a role account with the assigned rights profiles that contain the desired trusted applications. To access a role workspace, you must log in as a normal user, assume a role using the Trusted Path menu (or by clicking the role workspace button in the Front Panel if it already exists), and supply the role password. Note that the default label for a role workspace is the role's minimum label, usually ADMIN_LOW. If desired, you can switch labels by choosing `Change Workspace Label` from the Trusted Path menu while the pointer is over the role workspace button. To leave a role workspace temporarily, click any other workspace button. To destroy the workspace, choose `Delete` from the Trusted Path menu while the pointer is over the role workspace button.

Within the role workspace, you can access four types of trusted applications:

- Solaris Management Console tools—The Solaris Management Console (SMC) serves as a launcher for various administration tools and is available from: (1) the Application Manager, (2) the Tools subpanel in the Front Panel, and (3) the command line by typing `smc`.
- commands—In the Trusted Solaris environment, administrative commands and other commands intended for restricted use are assigned to rights profiles. Opening a terminal in a role workspace launches a profile shell that gives you access to all commands assigned to the account's rights profile(s). Any commands you run are at the label of the current workspace.
- CDE actions—The `System_Admin` folder in the Application Manager provides actions for performing miscellaneous system administration tasks. Most of these actions apply a special version of the `vi` editor, `adminvi(1M)` (or the `dtpad` editor if you prefer), to one of the configuration files. For security purposes, the editing actions cannot save a file to a different name, create a new file, or escape to a shell. All actions conform with mandatory access control and the local security policy. Any actions you launch are at the label of the current workspace (unless overridden by a rights profile).
- enhanced desktop tools—The Trusted Solaris operating environment provides desktop tools for administrators from the Front Panel that have capabilities not available to normal users. For example, the File Manager lets administrators set privileges and labels on executable files. Similarly, the Device Manager makes device administration capabilities available to roles. See “How the Trusted Solaris Environment Controls Device Access” on page 39.

Remote Administration

You can perform remote administration in the Trusted Solaris operating environment using the Solaris Management Console. You can also log into a remote host from another Trusted Solaris host in the system. Depending on your site's security policy, you can make adjustments to log in from a non-Trusted Solaris system, although this will make your system somewhat less secure. See “Administering Remote Systems” in *Trusted Solaris Administrator's Procedures*.

Solaris Management Console Tools

The Solaris Management Console (SMC) provides access to families of GUI-based administration tools. These tools let you edit items in various configuration databases.

SMC Toolboxes

The SMC tools are stored in collections referred to as *toolboxes*. For the security-related tools in the Trusted Solaris environment, you need to open the

toolbox called the Trusted Solaris Management Console. Within the Trusted Solaris toolbox, you can access tools according to scope, that is, the name service for the administration files accessed by the tools: local host, NIS, or NIS+.

Organization of the Solaris Management Console

The SMC is shown in the following figure, with the Trusted Solaris toolbox loaded and the User Tool open.

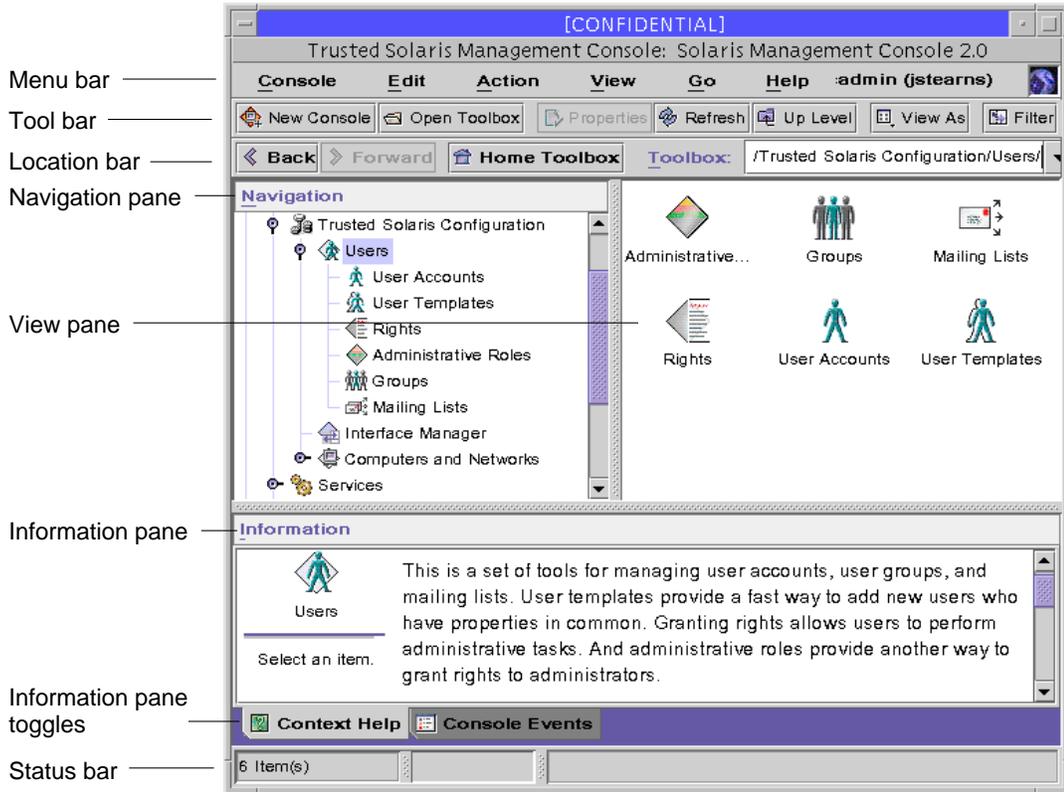


Figure 2-1 Typical Trusted Solaris SMC

At the top of the SMC there is a menu bar, a tool bar, and a location bar. At the bottom is the status bar. The status bar indicates the number of items in the navigation pane (at the left). The middle panel in the status bar is an indicator that a task is in progress and the right panel displays messages describing the current phase of the task.

The main part of the SMC consists of three panes:

- Navigation pane (at the left)—For accessing tools (or sets of tools), folders, or other toolboxes. Icons in the navigation pane are called nodes and are expandable

if they are folders or toolboxes. In this example, the Trusted Solaris Management Console toolbox icon is expanded; it contains the User Tool collection, the Interface Manager Tool, and the Computers and Networks Tool collection. The User Tool collection is selected and expanded also.

- View pane (at the right)—For viewing information related to the node selected in the navigation pane, either the contents of the selected folder, subordinate tools, or data associated with the selected tool. In this example, it displays the contents of the User Tool collection (which is also expanded in the navigation pane). Note that you can double-click a node in either the view pane or the navigation pane to open it.
- Information pane (at the bottom)—For displaying context-sensitive help or console events.

Changing the SMC Window

The layout of the SMC window is highly configurable. Use the following to change your layout:

- View menu—The Show option in the View menu hides or displays the optional bars and panes. The other options in the View menu control the display of nodes in the view pane.
- Console menu—The Preferences option lets you set: the initial toolbox, the orientation of panes, clicking or double-clicking for selection, text and/or icons in the tool bar, fonts, default tool loading, authentication prompts, and advanced logins.
- Context Help/Console Events toggles—The icons at the bottom of the information pane let you toggle between displaying context-sensitive help and console events.

SMC Documentation

The main source of documentation for using the SMC and its tools is the online help system. There are two forms of online help: context-sensitive help and expanded help topics. The context-sensitive help is tied to the currently selected feature and is displayed in the information pane. The expanded help topics are available from the Help menu or by clicking cross reference links in the context-sensitive help; the help topics appear in a separate viewer.

How SMC Tools Work

The SMC tools let you edit the attributes (referred to as *properties*) of items in the system databases. Interaction with the SMC tools take three general forms:

- Simple dialog boxes with online help on the left and data entry fields on the right. The Interface Manager below is an example; all its data can be displayed in the dialog box without the need for tabs.

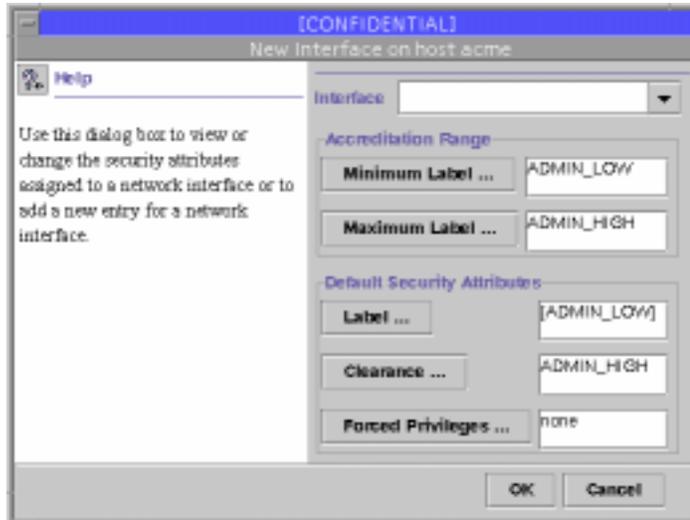


Figure 2-2 Simple SMC Tool Example

- Tabbed dialog boxes are used to edit large sets of attributes. The dialog boxes display online help on the left and data entry fields on the right. If there is more data than will fit in a single window, a file folder metaphor is used with selectable tabs at the top for choosing a category of data. Within each tab, data may be typed in directly, selected from a menu, or entered in a separate special-purpose dialog box. The User Manager below is an example of a tabbed dialog box.



Figure 2-3 Tabbed SMC Tool Example

- Wizards are series of dialog boxes for creating new data records. They take you through a series of steps to enter the new data. They have instructions built into the interface and use Next and Back buttons to progress through the series. Note that some wizards enter a subset of the data with the remainder being supplied as defaults; in such cases, you edit any changes in the corresponding properties dialog box. A typical example is the Add New User wizard below.

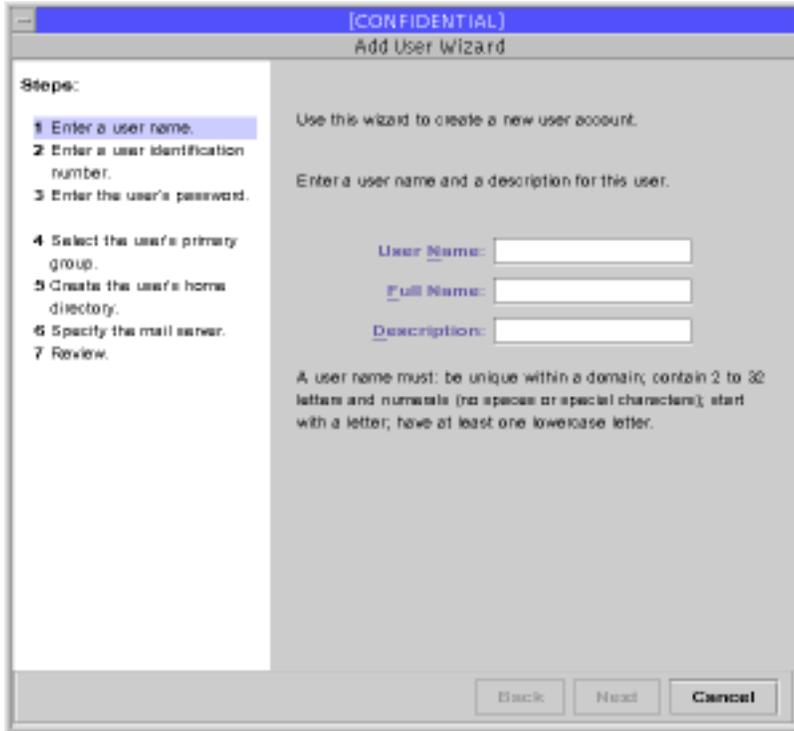


Figure 2-4 SMC Wizard Example

As a general rule, you open the tools either by selecting the tool icon (in the navigation pane or view pane) and choosing Open from the Actions menu or simply by double-clicking the icon. This will display icons representing data items in the view pane. The operations you can perform on data items are accessed through either the Actions menu or the popup menu, which is displayed by holding down the right mouse button.

Trusted CDE Actions

This section presents the CDE actions available to roles and describes how to use or change the restricted editor used in these actions. The trusted CDE actions are listed in the following table.

TABLE 2-1 Administrative Actions, Purposes, and Default Roles

Action Name	Purpose of Action	Default Rights Profile
Add Allocatable Device	Creates entries in <code>device_allocate(4)</code> , and <code>device_maps(4)</code> , and creates an auxiliary file for a new allocatable or nonallocatable device. User enters device name, device type, and lists all device special files associated with the device. See <code>add_allocatable(1M)</code> .	Device Security
Admin Editor	Edits any specified file	Object Access Management
Audit Classes	Edits <code>audit_class(4)</code>	Audit Control
Audit Control	Edits <code>audit_control(4)</code>	Audit Control
Audit Events	Edits <code>audit_event(4)</code>	Audit Control
Audit Startup	Edits the <code>audit_startup.sh</code> script [see <code>audit_startup(1M)</code>]	Audit Control
Check Encodings	Runs <code>chk_encodings(1M)</code> on specified encodings file	Object Label Management
Check TN Files	Runs <code>tnchkdb(1M)</code> on local <code>tnidb(4)</code> , <code>tnrhdb(4)</code> , and <code>tnrhtp(4)</code> files	Network Security
Check TN NIS+ Tables	Runs <code>tnchkdb(1M)</code> on <code>tnrhdb(4)</code> , and <code>tnrhtp(4)</code> NIS+ trusted network maps	Network Management
Configure Selection Confirmation	Edits <code>/usr/dt/config/sel_config</code> [see <code>sel_config(4)</code>]	Object Label Management
Create NIS Client	Runs <code>ypinit(1M)</code> , using both the specified hostname for the NIS master and the specified domain name	Name Server Security
Create NIS+ Client	Runs <code>nisclient(1M)</code> , using both the specified hostname for the NIS+ master and the specified domain name	Name Server Security
Create NIS Server	Runs <code>ypinit(1M)</code> using the specified domain name	Name Server Security
Create NIS+ Server	Runs <code>nissserver(1M)</code> using the specified domain name	Name Server Security

TABLE 2-1 Administrative Actions, Purposes, and Default Roles *(continued)*

Action Name	Purpose of Action	Default Rights Profile
Edit Encodings	Edits specified <code>label_encodings(4)</code> file and runs <code>chk_encodings(1M)</code>	Object Label Management
Name Service Switch	Edits <code>nsswitch.conf(4)</code>	Network Management
Populate NIS Tables	Runs <code>nispopulate(1M)</code> from the specified directory	Name Service Security
Set Daily Message	Edits <code>/etc/motd</code>	Network Management
Set Default Routes	Edits <code>/etc/defaultrouter</code> [see the <code>route(1M)</code> man page]	Network Management
Set DNS Servers	Edits <code>resolv.conf(4)</code>	Network Management
Set Mail Options	Edits <code>/etc/mail/sendmail.cf</code> [see <code>sendmail(1M)</code>]	Mail Management
Set Mount Attributes	Edits <code>vfstab_adjunct(4)</code>	File System Security
Set Mount Points	Edits <code>vfstab(4)</code>	File System Management
Set Tsol Gateways	Edits <code>tsolgateways(4)</code>	Network Management
Shared Filesystem	Edits <code>dfstab(4)</code> ; does not run <code>share(1M)</code>	File System Management
View Table Attributes	Runs <code>niscat(1)</code> with the <code>-o</code> option on the specified NIS+ trusted network database to display the table's attributes.	Name Service Management
View Table Contents	Runs <code>niscat(1)</code> on the specified NIS+ trusted network database to display the table's contents.	Name Service Management

Admin Editor

The Admin Editor action, which can also be accessed from the command `adminvi(1M)` is a modified version of the `vi(1)` command. It is restricted to prevent

the user from executing shell commands and from writing to (saving to) any file other than the original file being edited. The `Admin Editor` action, which is assigned to the security administrator role by default, should be used in most cases instead of `adminvi` on the command line to edit or create administrative files. (This is due to the fact that the `Admin Editor` is wrapper for `adminvi` that incorporates auditing and allows an editor preference.) You can assign the `adminvi` command to any users with the profile shell as their default if you need to provide them a text editor with the restrictions of `adminvi`.

Changing the Default Admin Editor

The admin editor is launched through the `/usr/dt/bin/trusted_edit` shell script, which brings up the editor specified in the `EDITOR` environment variable for the role account, restricts saves, and audits any changes made at the time the file is saved. The variable is set to `adminvi(1M)` by default, but the security administrator role can redefine the `EDITOR` variable to `/usr/dt/bin/dtpad`. When `adminvi` is specified, `/bin/adminvi` is invoked as root to edit the file. The `adminvi` command prevents the saving of the file with any other name. If `dtpad(1)` is specified, the `New`, `Save`, and `Open` options in the `File` menu are disabled when the action runs, so that the file cannot be renamed.

Administering Users

You can administer users through either the SMC User Tool applications or from the command line. This section is divided into these parts:

- “User Attribute Databases” on page 57
- “User Properties Dialog Box” on page 60
- “Right Properties Dialog Box” on page 62

Note - To administer users, you need the User Manager rights profile (for general user attributes) and the User Security rights profile (for security-related attributes).

Default User Attributes

The task of entering new users is greatly simplified by setting up default user attributes so that only those attributes unique to a specific user need be added. There are three mechanisms for setting up defaults:

- `policy.conf(4)` database—lets you specify authorizations, rights profiles, password generation, account locking, label display, and unattended workstation controls.
- `label_encodings(4)` database—lets you specify default values for user clearances and minimum SLs and public alternative names for `ADMIN_HIGH` and `ADMIN_LOW`.
- user templates—let you specify all user properties not covered by the `policy.conf(4)` and the `label_encodings(4)` databases except properties specific to a user such as user name and ID.

The tools for creating new users are the `Add User With Wizard...` and `Add User From Template...` menu options. The wizard approach offers simplicity but with these tradeoffs:

- The login shell defaults to Bourne.
- It does not set a skeleton path for initialization files.
- Secondary groups are not set.

The user template approach offers a larger set of user properties, but requires you to set up one or more templates of default user attributes ahead of time. Both methods should be used in conjunction with the `policy.conf(4)` and the `label_encodings(4)` databases. The User Properties dialog box lets you make modifications after the initial user information has been entered.

User Attribute Databases

The user information is held in the following databases:

- `user_attr(4)`—The `/etc/user_attr` file contains extended user attributes, using a `keyword=value` format.
- `auth_attr(4)`—The `/etc/security/auth_attr` file contains the definitions of authorizations, which can be included in rights profiles.
- `prof_attr(4)`—The `/etc/security/prof_attr` file contains the name, description, authorizations, subordinate rights profiles, and help files for rights profiles.
- `exec_attr(4)`—The `/etc/security/exec_attr` file contains commands and actions with security attributes assigned to rights profiles.

These databases can be edited manually, although this practice is not generally recommended.

The following figure shows how the databases work together to provide user attributes.

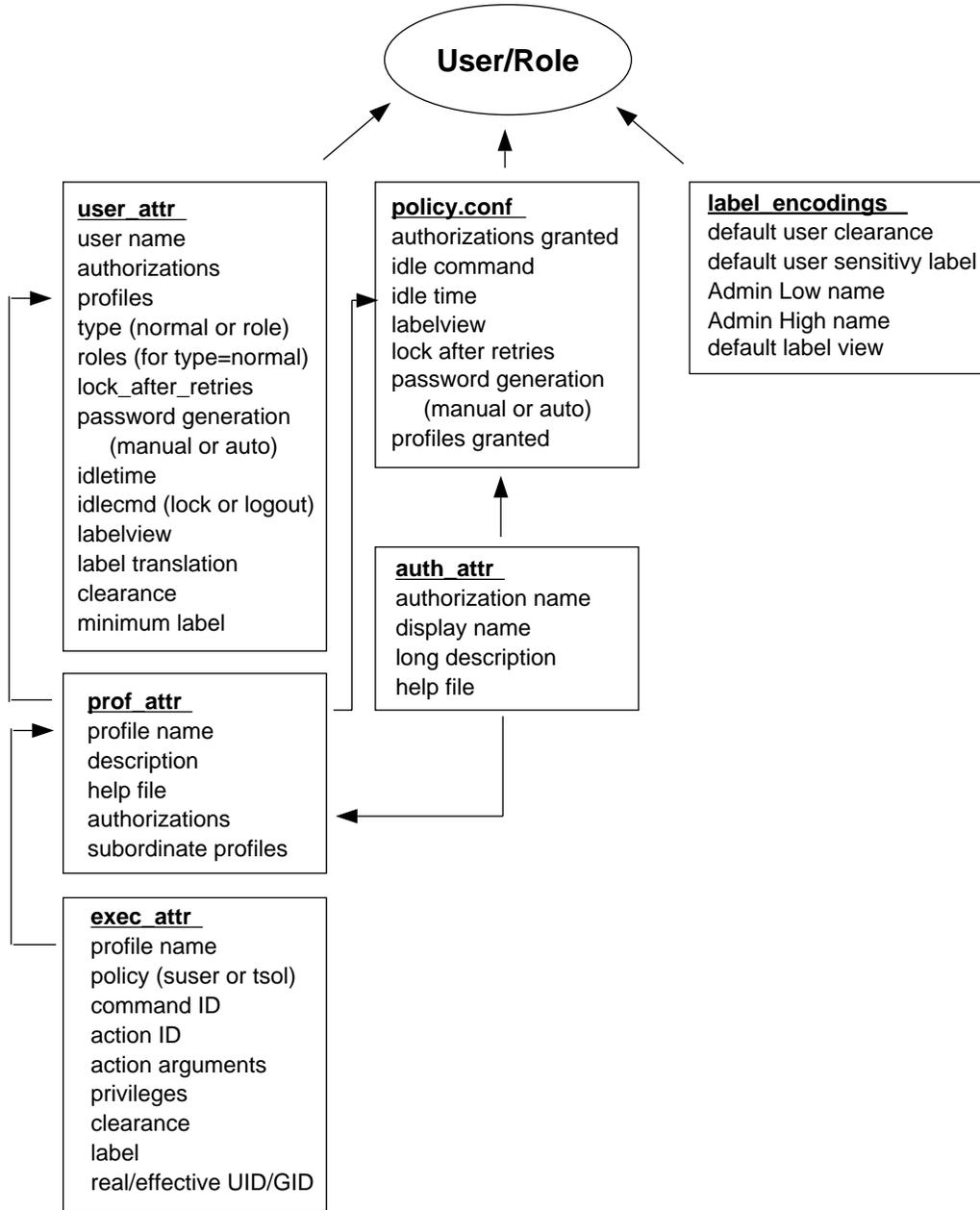


Figure 2-5 User Database Relationships

The `user_attr` database contains the attributes shown, including a comma-separated list of profile names. The contents of the profiles are split between the `prof_attr` file, which contains profile identification information, authorizations assigned to the profile, and subordinate profiles, and the `exec_attr` file, which

contains commands and actions with their associated security attributes. The `auth_attr` file supplies available authorizations to the `prof_attr` file and the `policy.conf` file. (Note that although you can assign authorizations directly to users through `user_attr`, this practice is discouraged.) The `policy.conf` file supplies default attributes to be applied to all users. The `label_encodings` file supplies label defaults if they are not otherwise specified.

Managing Users from the Command Line

The user files can also be managed from the command line. The `smuser(1M)` command adds, modifies, deletes, and lists user information. You can use `smmultiuser(1M)` to enter a batch of users.

Managing Users through the SMC

This section describes the SMC User Tool collection and selected dialog boxes as follows:

- “User Tool Collection Summary” on page 59
- “User Properties Dialog Box” on page 60
- “Right Properties Dialog Box” on page 62

For complete descriptions of elements in the User Tool collection, refer to the online help.

User Tool Collection Summary

The SMC User Tool collection is shown in the following figure.



Figure 2-6 SMC User Tool Collection

The six dialog boxes in the User Tool collection are:

- Administrative Roles dialog box—Lets you create or edit a role account and assign users to roles. Note that the roles data is the same as the user data except that (1)

there is no Roles tab since roles cannot be assigned to other roles, (2) there is no Password Options tab because these are not appropriate for roles, and (3) the Roles dialog box has a Users tab for assigning users to the role.

- Groups dialog box—Lets you create or edit user groups and change the members in the group.
- Mailing Lists dialog box—Lets you create or edit mail aliases, including changing the recipients in the list.
- Rights dialog box—Lets you create or edit a rights profile. See “Right Properties Dialog Box” on page 62 for an example of the Rights Properties dialog box and a description of the rights profile data.
- User Accounts dialog box—Lets you add new users singly or in a batch, with or without a template, and lets you edit the properties of existing users. See “User Properties Dialog Box” on page 60 for an example of the User Properties dialog box and a description of the user data.
- User Templates dialog box—Lets you create a named set of user properties that can be applied to new users to facilitate data entry.

User Properties Dialog Box

The User Properties dialog box is shown below with the General tab selected.



Figure 2-7 User Properties Dialog Box

The following table describes the purpose of each tab in the User Properties dialog box.

TABLE 2-2 User Properties Summary

Tab	Description
General	Specifies the user, the default login shell, and the account availability.
Group	Sets the user's primary and secondary groups for the purpose of accessing and creating files and directories.
Home Directory	Specifies the user's home directory, home directory server, automounting, and directory access.
Password	Specifies whether the user or the administrator will select the first password and whether the selection and changes will be manual or from the password generator.

TABLE 2-2 User Properties Summary *(continued)*

Tab	Description
Password Options	Sets the time limits and requirements for password changes.
Mail	Specifies the server that provides email and the mailbox in which it is received.
Rights	Allows rights profiles to be assigned to the user. The precedence of the assigned rights profiles can be changed.
Roles	Allows available roles to be assigned to the user.
Trusted Solaris Attributes	Specifies the clearance and minimum label at which the user can operate and how labels are displayed to the user. Also specifies a time limit for which a workstation may remain idle and the action taken when the limit is reached.
Audit	Specifies the audit classes for which the user is to be audited.

Right Properties Dialog Box

The Rights Properties dialog box is shown below with the General tab selected.



Figure 2-8 Rights Properties Dialog Box

The following table describes the purpose of each tab in the Right Properties dialog box

TABLE 2-3 Rights Manager Dialog Box Summary

Tab	Description
General	Identifies and describes the rights profile and provides the name of the help file used to explain it.
Commands	Assigns commands to the rights profile and adds security attributes (effective and real UIDs and GIDs; minimum label and clearance; and inheritable privileges) to specific commands in the profile.
Actions	Assigns CDE actions to the rights profile and adds security attributes (effective and real UIDs and GIDs; minimum label and clearance; and inheritable privileges) to specific actions in the profile.

TABLE 2-3 Rights Manager Dialog Box Summary *(continued)*

Tab	Description
Authorizations	Assigns authorizations to the profile.
Supplementary Rights	Specifies other rights profiles to be contained within the current rights profile.

Administering Hosts and Networks

To administer hosts and networks, you need to open the Computers and Networks tool collection. A typical collection is shown in the view pane of the following figure.

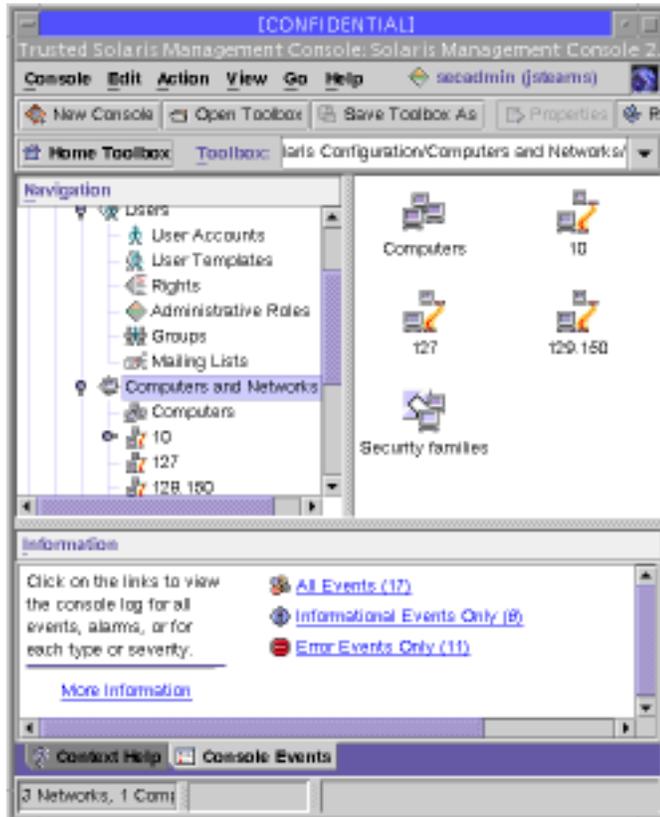


Figure 2-9 Computers and Networks Tool Collection

This gives you access to three tools:

- **Computers tool**—When open, the host icons for all local networks are displayed in the view pane, which let you edit IP address, ethernet address, and host alias information. A typical Computer properties dialog box is shown below.

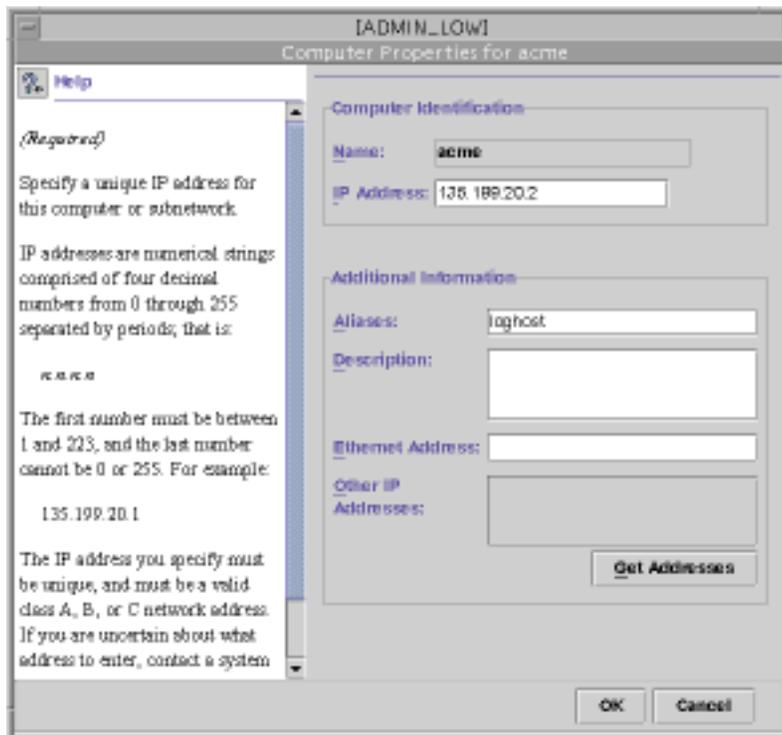


Figure 2-10 Computer Properties Dialog Box

- Subnetwork tool—This tool groups hosts by subnetwork and works the same as the Computers tool above. Its icon is displayed as two monitors connected by a cable, with a partial IP address as a caption.
- Security families tool—Lets you add or modify network templates including the assignments of hosts to the templates. This tool is described in more detail below.

Security Families Tool Set

A *security family* is a group of workstations that use a common networking protocol and have the same security requirements. As a result, you can apply the same template of network security attributes to them for the purpose of receiving and transmitting data. Trusted networking and templates are explained in more detail in Chapter 3.

When the Security Families tool is opened, all available templates display as icons. You can modify either the templates or the host assignments as follows:

- If you double-click a template icon, all hosts in that security family, that is, those assigned to the selected template, are displayed as icons. Double-clicking a host

icon (or selecting it and choosing Properties from the Action or popup menu) lets you modify its IP address or template assignment.

- If you select a template icon and choose Properties from the Action or popup menu, the Modify Template dialog box is displayed, as illustrated in the following figure, and you can change the definition of the template.

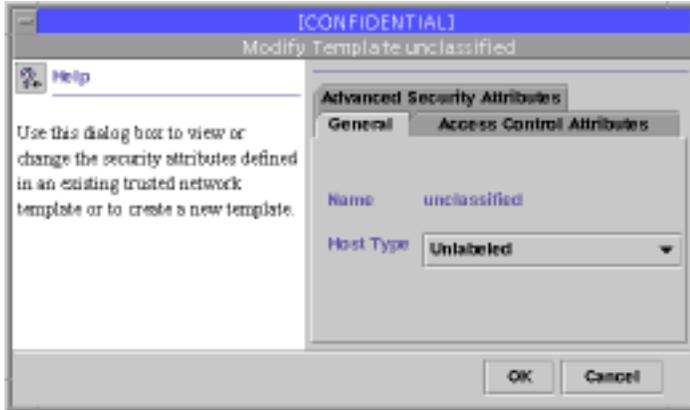


Figure 2-11 Modify Template Dialog Box

The tabs in the Modify Template dialog box are described in the following table.

TABLE 2-4 Template Dialog Box Summary

Tab	Description
General	Specifies templates, host types, and minimum/maximum labels.
Access Control Attributes	Specifies security attributes to be applied to incoming data from hosts to which this template is applied. The potential incoming security attributes include minimum label, maximum label, default label, and default clearance.
Advanced Security Attributes	Specifies security attributes to be applied to outgoing data to hosts to which this template is applied. The potential outgoing security attributes include DOI, IP label type, forced privileges, allowed privileges, RIPS0 send class, RIPS0 send PAF, RIPS0 return PAF, and CIPS0 domain.

Administering Other Aspects of the Trusted Solaris Environment

This section lists other commands available for administering elements in the Trusted Solaris operating environment.

File Management Commands

File privileges and labels can be administered either through the File Manager or the following commands:

- `getfattrflag(1)`—for getting a file's security attributes.
- `setfattrflag(1)`—for setting a file's security attributes.
- `getfpriv(1)`—for getting an executable file's forced and allowed privileges.
- `setfpriv(1)`—for setting an executable file's forced and allowed privileges.
- `getlabel(1)`—for getting a file's label.
- `setlabel(1)`—for setting a file's label.
- `testfpriv(1)`—for checking an executable file's forced and allowed privilege sets.

File System Management Commands

The following commands are for administering attributes on file systems.

- `getfsattr(1M)`—for displaying the security attributes of a file system.
- `getfsattr_ufs(1M)`—for displaying the security attributes of a UFS file system.
- `setfsattr(1M)`—for setting the security attributes on a file system. The file system should be unmounted first.
- `newsecfs(1M)`—for setting security attributes on a new file system.

Mount Management

The following commands are for mounting file systems. Check the Trusted Solaris Summary section of each man page for differences from the Solaris operating environment.

- `mount(1M)`—requires the `sys_mount` privilege. Both mandatory and discretionary read access (or overriding privileges) are required to the mount point and the

device being mounted. Depending on the configuration of the `vfstab_adjunct` file, the process may need some combination of the `proc_setsl` and `proc_setclr` privileges. The `mount` command supports mounts to multilabel directories (MLDs). It has a special option, `--S` which lets you specify security attributes to be associated with the filesystem mount (this option requires that you have sufficient clearance for the label specified).

- `share_nfs(1M)`—provides these options with `-S`:
 - `dev|nodev` – access to character and block devices is allowed or disallowed. The default is `dev`.
 - `priv|nopriv` – Forced privileges on execution are allowed or disallowed. The default is `priv`.

Running `share_nfs` requires the following:

 - `sys_nfs` privilege
 - effective uid 0
 - process label of `[ADMIN_LOW]`
- `share(1M)`—makes a resource of a specified file system type available for mounting. It requires the `sys_nfs` privilege.
- `unshare(1M)`—makes a resource unavailable for mounting. It requires the `sys_nfs` privilege.
- `nfsstat(1M)`—lets you display statistics concerning the NFS and RPC (remote procedure call) interfaces to the kernel. The Trusted Solaris version of the `nfsstat` command requires that you have the `net_config` privilege when using the `-z` option, which reinitializes the statistics.
- `nfsd(1M)`—handles client file system requests. The Trusted Solaris version of the `nfsd` command requires the `sys_nfs` and `net_mac_read` privileges to run.

Process Commands

The following commands are for managing processes:

- `pattr(1)`—lets you display the viewable Process Attribute Flags of the current process or a process specified by `pid`. Those flags that cannot be viewed normally can be viewed with privilege.
- `pclear(1)`—lets you display the clearance at which the selected process is running.
- `plabel(1)`—gets the CMW label (that is, combined sensitivity label and information label) for the process.
- `ppriv(1)`—gets the effective privileges of a process.
- `pprivtest(1)`—tests if the specified privileges are currently in effect.

Administering Trusted Networking

This chapter describes networking in the Trusted Solaris environment. The Trusted Solaris networking subsystem is an enhanced version of the regular Solaris TCP/IP network. The extensions enable communication between workstations on the network in a trusted fashion. The networking subsystem helps ensure that the system's security policy (for example, MAC) is preserved across distributed applications. The amount of administration and protection required for your network depends on whether it is homogeneous or heterogeneous.

Note - In the default configuration, the security administrator role is responsible for network security.

- “Overview of Trusted Solaris Networking” on page 71
- “Routing in Trusted Solaris” on page 78
- “Trusted Solaris Network Commands” on page 83
- “Troubleshooting Networks” on page 84

Overview of Trusted Solaris Networking

This section covers the following networking topics:

- Homogeneous networks
- Heterogeneous networks
- Host types
- Network configuration databases
- Related subsystems

- How data is transmitted

Homogeneous Networks

A homogeneous network configuration is the easiest to administer and protect. In a *homogeneous network configuration*, all workstations run the Trusted Solaris operating environment and use the same NIS or NIS+ master server with the same set of security attributes (clearances, labels, etc.). A typical homogeneous network, served by a NIS+ master, is shown in the following figure. The hosts in a homogeneous network are said to be in the same *security domain*.

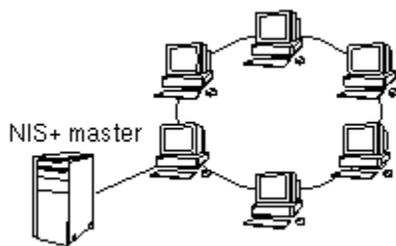


Figure 3-1 Homogeneous Network

Workstations are connected to networks by a physical connector called a *network interface*. Each network interface has an accreditation range, consisting of a maximum label setting the upper boundary and a minimum label for the lower boundary. The accreditation range controls the sensitivity of the information that can be transmitted or received through the interface.

A single computer running the Trusted Solaris operating environment by itself is considered to be a standalone security domain.

Heterogeneous Networks

Trusted Solaris networks can also accommodate hosts running different network protocols. A heterogeneous configuration requires more administration than a homogeneous arrangement; you must specify how data from hosts with different protocols will be treated with regard to security policy. The following figure shows a typical heterogeneous network and some different protocols with which a Trusted Solaris network can communicate.

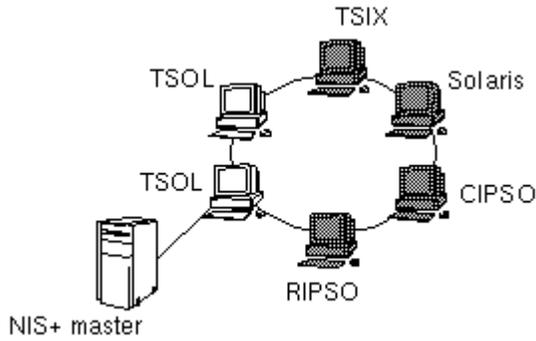


Figure 3-2 Heterogeneous Network

Trusted Solaris Data Packets

To understand how Trusted Solaris workstations accept data from other Trusted Solaris workstations and hosts using other data protocols, it is useful to compare the standard data packet formats with the Trusted Solaris formats (see figure below).

(a) Standard IPv4 packet

IPv4 header w/ options	TCP or UDP	Data
------------------------	------------	------

(b) Trusted Solaris IPv4 packet

IPv4 header w/ options	TCP or UDP	SAMP	Data
------------------------	------------	------	------

(c) Standard IPv6 packet

IPv6 header w/ extensions	TCP or UDP	Data
---------------------------	------------	------

(d) Trusted Solaris IPv6 packet

IPv6 header w/ extensions (MLS option)	TCP or UDP	Data
--	------------	------

Figure 3-3 Comparison of Data Packet Formats

In the standard IPv4 format, there is a header with options, followed by a TCP or UDP header and the actual data. The Trusted Solaris version of an IPv4 packet uses the IP options in the header for security attributes and also a SAMP (Security Attribute Modulation Protocol) header identifying the session management protocol and version and security attributes.

The standard IPv6 format contains a header with extensions, followed by a TCP or UDP header and the actual data. The Trusted Solaris IPv6 packet includes a multilevel security option in the header extensions.

When you configure the network configuration databases for your site, you specify all hosts with which workstations on your network can communicate. You set up templates with default security attribute values, categorized by the host types as explained in the following section.

Security Families

Network administration in the Trusted Solaris environment is based on the concept of security families, that is, treating host machines with common protocols and identical security requirements the same way. For a host to be able to communicate with other hosts on a Trusted Solaris network, you must identify its host type, that is, its networking protocol, and assign it a template of security attributes.

Host Types in Networking

Trusted Solaris classifies host types according to the networking protocols as follows:

- **Trusted Solaris**—refers to workstations running Trusted Solaris. It uses binary representation for security attributes in the protocol.
- **unlabeled**—refers to hosts that do not send or recognize security attributes.
- **TSIX**—refers to hosts supporting the TSIX (RE) 1.1 (Trusted Systems Information eXchange for Restricted Environments standard). It uses the same format as Trusted Solaris hosts (see Figure 3-3) except that it uses tokens (arbitrary 32-bit numbers) rather than binary data to represent security attributes. The tokens use the security attribute token mapping protocol (SATMP).
- **CIPSO**—refers to hosts conforming to CIPSO, TSIX (RE) 1.1. The only security attributes supported under CIPSO are the DOI (domain of interpretation) and CIPSO label.
- **RIPSO**—refers to hosts conforming to RIPSO, as described in the IETF RFC 1108. The Trusted Solaris environment supports an administratively-set fixed RIPSO label to be applied to incoming and outgoing network packets. Although this functionality does not fully meet the RFC specifications, it supplies sufficient functionality where RIPSO labels are needed.

Note - The TSIX, CIPSO, and RIPSO host types lie in the category of hosts running other trusted operating environments. The unlabeled host type is intended for those hosts that use the standard networking protocol and do not support security attributes.

Networking Security Attributes

The security attributes that can be specified in networking templates are:

- minimum label—defines the bottom of the label range for this security family. Outgoing packets to hosts in this security family cannot be below the minimum label.
- maximum label—defines the top of the label range for this security family. Outgoing packets to hosts in this security family cannot be higher than the maximum label.
- default label—sets the label to be applied by default to incoming packets from hosts in this security family.
- default clearance—sets the clearance to be applied by default to incoming packets from hosts in this security family.
- DOI—an integer that identifies the domain of interpretation, that is, the labelling scheme used by the default label and clearance for the particular host type.
- IP label—identifies type of IP label: RIPS0, CIPS0, or none. If CIPS0, the `/etc/system` and `label_encodings` files must be modified to accommodate the ADMIN_HIGH label (see the “About Security Families” help card). If RIPS0, you must specify a RIPS0 label for the RIPS0 Send Class
- allowed privileges—can be used to restrict privileges available to remote Trusted Solaris hosts. If these hosts can use any privileges, set `ALL`; if there is a limit, specify only those privileges that can be applied.
- forced privileges—sets privileges to enable a remote host, typically an unlabeled host, to perform specific functions that may override security policy.
- RIPS0 Send Class—used by RIPS0 hosts and with RIPS0 IP labels only, the classification level at which datagrams sent to a host of that template are protected. The predefined Classes are Top Secret, Secret, Confidential and Classified.
- RIPS0 Send PAF (protection authority flag)—used by RIPS0 hosts and with RIPS0 IP labels only, the bit mask identifying the protection authorities on datagrams sent to a host of that template. The predefined authorities are: GENSER, SIOP-ESIm SCI, NSA, and DOE.
- RIPS0 Return PAF (protection authority flag)—used by RIPS0 hosts and with RIPS0 IP labels only, specifies the PAF portion of the RIPS0 label on ICMP error messages sent back from hosts using this template.

Networking Templates

The purpose of the Trusted Solaris networking templates is to specify the security attribute values to be applied to hosts within a security family. Not all of the security attributes are appropriate to each host type. The following table indicates how

security attributes are applied to which host types. The term *default* means that the attribute is supplied by default. *Optional* means that is your choice whether to use this default. *Not allowed* means that any entry will be ignored. *Required* with or without conditions means the attribute is mandatory.

TABLE 3-1 Security Attributes by Host Type

Host Types —> Security Attributes	Trusted Solaris	TSIX	Unlabeled	CIPSO	RIPSO
minimum label	default	default	default	default	default
maximum label	default	default	default	default	default
default label	not allowed	not allowed	default	not allowed	default
default clearance	not allowed	not allowed	default	default	default
DOI	optional	optional	optional	optional	optional
IP label	optional	optional	optional	optional	optional
forced privileges	not allowed	not allowed	default	default	default
allowed privileges	default	default	not allowed	not allowed	not allowed
RIPSO Send Class	required if host or IP label is RIPSO	not allowed	required if host or IP label is RIPSO	not allowed	required
RIPSO Send PAF	required if host or IP label is RIPSO	not allowed	required if host or IP label is RIPSO	not allowed	required
RIPSO Return PAF	required if host or IP label is RIPSO	not allowed	required if host or IP label is RIPSO	not allowed	required

Network Configuration Databases

There are three network configuration databases for establishing external communication:

- `tnrhdb`
- `tnrhtp`
- `tnidb`

These databases are loaded into the kernel and are used in accreditation checks as data is transmitted from one host to another. These databases are maintained using the Computers and Security Families dialog boxes in the SMC Computers and Networks tool and the SMC Interface Manager. Trusted Solaris can use a naming service for central management of the `tnrhdb` and `tnrhtp` databases; the `tnidb` database is maintained separately on each host.

Network host information is stored in the `tnrhdb(4)` database. It holds the IP addresses for all hosts permitted to communicate with workstations in the network and the templates (from the `tnrhtp` database) assigned to them. The `tnrhdb` database can also hold default values as part of a fallback mechanism. Substituting 0 in the rightmost byte(s) of the IP address serves as a wildcard for unlisted hosts with IP addresses that match the non-zero portion of the default. You can also set a fixed prefix length by adding a slash (/) followed by the number of fixed bits. See the following table for examples.

TABLE 3-2 `tnrhdb` Fallback Mechanisms Example

<code>tnrhdb</code> Entry	Addresses Covered
129.150.118.0:tsol	addresses beginning with 129.150.118.
129.150.0.0:tsol	addresses beginning with 129.150.
129.0.0.0:tsol	addresses beginning with 129.
0.0.0.0:tsol	all addresses on network
129.150.118.128/26:tsol	addresses from 129.150.118.0 to 129.150.118.63

Network template information is stored in the `tnrhtp(4)` database. In a homogeneous network, only one template is needed; in a heterogeneous network, you need a separate template for each type of host. The attributes in the templates provide attributes from incoming data. They also provide destination information for outgoing data and are used in accreditation checks for incoming packets.

The `tnidb(4)` database is local to each host. It contains the host's network interfaces with their accreditation ranges. Default values for labels, clearances, effective UIDs/GIDs, and forced privileges apply to communications to and from hosts running environments that do not support these attributes. Note that any default values set in `tnrntp` override the values in `tnidb`. By default, the file is empty because default values are used for all interfaces.

Related Subsystems

The trusted NFS feature of Trusted Solaris permits mounting between Trusted Solaris hosts and the other host types. Transmitted data is protected by MAC and DAC. Missing labels are supplied by the `tnrntp` and `tnidb` databases. For more information, see "Mounting Various Types of File Systems in the Trusted Solaris System" in *Trusted Solaris Administrator's Procedures*.

Routing in Trusted Solaris

In the Trusted Solaris operating environment, routes between hosts on different networks must maintain security at each step in the transmission.

Loading Routing Information at Boot Time

When a Trusted Solaris host boots, it loads routing information so it can transmit data. If the file `/etc/tsolgateways` (which is maintained manually by the administrator) exists, then the gateways in the file serve as the host's defaults. If, however, `/etc/tsolgateways` does not exist, then the host uses the default routes from the file `/etc/defaultrouter`, which is also maintained manually by the administrator. If either file exists, then the host is said to use *static routing*.

If neither the `/etc/tsolgateways` nor the `/etc/defaultrouter` file exists, then the host uses *dynamic routing* and must start a special daemon, either `in.rdisc(1M)` (the network router discovery daemon) or `in.routed(1M)` (the network routing daemon). If the host also serves as a gateway (that is, a host that connects to two or more networks), then both `in.rdisc` and `in.routed` are started.

Routing Tables in the Trusted Solaris Environment

The main objective for routing is to find the shortest secure route between two hosts. Trusted Solaris routing tables are based on extended metrics (called emetrics). An

emetric is a combination of a routing metric and Security Routing Information (SRI), for measuring security. The SRI can incorporate these security attributes:

- Minimum SL
- Maximum SL
- DOI
- RIPS0 label
- RIPS0 error
- CIPS0 only
- RIPS0 only

This information is propagated by the routing daemon `in.routed` using the Trusted Solaris-extended Routing Information Protocol if dynamic routing is used, or if static routing is used, by manual entry using the `route` command or through the `/etc/tsolgateways` or `/etc/defaultrouter` files. The *emetric* for a particular route is used for accreditation checks when this route is being considered.

Not every route in the routing table must have an *emetric*. If a route does not have an *emetric*, the remote host template of its first hop gateway is used for the accreditation check instead.

Accreditation Checking

To determine the suitability of a route regarding security, Trusted Solaris runs a series of tests called *accreditation checks* on the source host, destination host, and the route's *emetrics*. If the *emetric* for a particular route is missing, the security attributes for the first-hop gateway in the route are checked. A host's security attributes are derived from information in the `tnrhdb`, `tnrhtp`, and `tnidb` files. The tests check, for example, that a data packet's label is within the range of each host in the route.

Source Accreditation Checks

The accreditation checks conducted on the source host are:

- The label of the data being sent must be within the destination host's accreditation range.
- The label of the data must be within the accreditation range of the *emetric* for the route or, if the *emetric* is not available, first-hop gateway's security attributes.
- The label of the data must be within the accreditation range of the source host's network interface.
- The DOI of an outgoing packet must match the DOI of the destination and the route's *emetric* (or first-hop gateway).

- An outgoing packet's RIPS0 label must match the RIPS0 label of the destination and the route's emetric (or first-hop gateway). Alternatively, the RIPS0 error can match the destination's RIPS0 error, the route's emetric, or the first-hop gateway's RIPS0 error.

Gateway Accreditation Checks

The accreditation checks conducted on a Trusted Solaris gateway host are:

If the next hop is an unlabeled host, then the label of the source host must match the label of the destination host.

If the packet has the CIPSO option, the following conditions for forwarding must be true:

- The route's emetric (or next-hop gateway) must be able to accept data in the CIPSO protocol.
- The route's emetric (or next-hop gateway) must be in the data packet's DOI.
- The DOI (from the `tnrhtp` database) for the outgoing interface must be the same as the data packet's DOI.

If the packet has the RIPS0 option, the following conditions for forwarding must be true:

- The route's emetric (or next-hop gateway) must be able to accept data in the RIPS0 protocol.
- The route's emetric (or next-hop gateway) must have the same RIPS0 label (or RIPS0 error) as the data packet's RIPS0 label (or RIPS0 error).

Destination Accreditation Checks

When a Trusted Solaris machine receives data, the trusted network software checks for the following:

- The label of the data is within the accreditation range of both the source machine and the network interface receiving the data.
- If a packet has a CIPSO label, then the DOI in the packet must be the same as the DOI in the remote host template for the destination.
- If a packet has a RIPS0 label (or RIPS0 error), then the RIPS0 label (or RIPS0 error) in the packet must be the same as the RIPS0 label (or RIPS0 error) in the remote host template for the destination.

After the data has passed the accreditation checks above, the system checks that all necessary security attributes are present. If there are missing attributes, the system looks up the source host (by its IP address or a target expression) in the `tnrhdb` database to get the name of the network security template assigned to the host. The system then retrieves the template's set of security attributes from the `tnrhtp`

database. If there are still security attributes missing, the software looks up the network interface in the `tnidb` database and retrieves default security attributes.

Routing Example

An example of routing in the Trusted Solaris environment is shown in the following figure; Figure 3-4 (a) shows the routing diagram and Figure 3-4 (b) shows the routing table. There are three potential routes between Host 1 and Host 2:

- Route #1 is the shortest with a Routing Information Protocol (RIP) metric of 3. Datagrams using route #1 are restricted to a label range of CONFIDENTIAL (C) to SECRET (S).
- Route #2 has a larger label range of ADMIN_LOW to ADMIN_HIGH. Datagrams using route #2 must use have an IP Option set to CIPSO.
- Route #3 has the longest distance of the three routes with an RIP of 6. Its Security Routing Information is unknown, so any security attributes must be derived from the template in `tnrhtp` for Gateway #5.

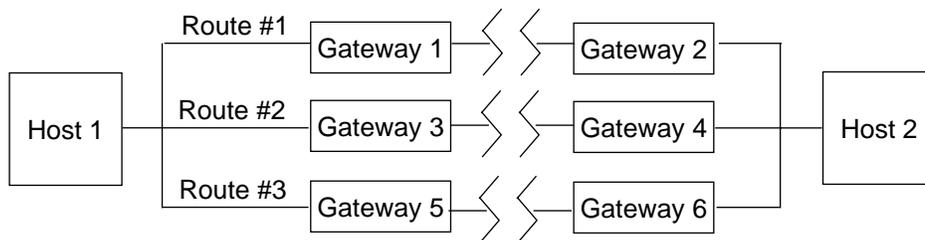


Figure 3-4 Typical Trusted Solaris Routes and Routing Table

Route	First hop gateway	RIP Metric	Min SL	Max SL	DOI	CIPSO
1	Gateway 1	3	C	S		
2	Gateway 3	4	ADMIN_LOW	ADMIN_HIGH		Y
3	Gateway 5	6				

Using Routing Commands

To display the contents of the routing table, use the command `netstat` with the `-R` option. To make a manual change to the routing table, use the `route` command with the `add` or `delete` option. For example,

```
% route add net 129.150.115.0 129.150.118.39 \
-m metric=2,min_sl=c,max_sl=ts,ripso_label="top_secret_sci",\
ripso_error="genser;sci"
```

```
add net 129.150.115.0: gateway 129.150.118.39
```

adds to the routing table a loop with the hosts at 129.150.115.0 and 129.150.118.39 with a distance metric of 2, an SL range from C to TS, a RIPS0 label = top_secret sci, and a RIPS0 error = genser;sci. To see the results of the added loop, type:

```
% netstat -Rn
...
129.150.115.0      129.150.118.39      UG      0      0
                  metric=2,min_sl=C,max_sl=TS,ripso_label=0x3d 0x20000000 (top_secret sci)
,ripso_error=0xa0000000 (genser;sci)
...
```

The new route is shown above. The other routes are replaced by ellipses (...). A second example of adding a route with two new emetrics and viewing the new routing table follows:

```
% route add net 129.150.114.0 129.150.118.39 \
-m metric=3,min_sl=admin_low,max_sl=s,doi=3 \
-m metric=4,min_sl=c,max_sl=admin_high,doi=4,ripso_label="top_secret sci",\
ripso_error="genser;sci"

add net 129.150.114.0: gateway 129.150.118.39
% netstat -Rn
...
129.150.115.0      129.150.118.39      UG      0      0
                  metric=2,min_sl=C,max_sl=TS,ripso_label=0x3d 0x20000000 (top_secret sci)
,ripso_error=0xa0000000 (genser;sci)
129.150.114.0      129.150.118.39      UG      0      0
                  metric=4,min_sl=C,max_sl=ADMIN_HIGH,doi=4,ripso_label=0x3d 0x20000000 (t
op_secret sci),ripso_error=0xa0000000 (genser;sci)
                  metric=3,min_sl=ADMIN_LOW,max_sl=S,doi=3
...
```

Routing through Non-Trusted Solaris Gateway Clusters

It is possible to route secure data through clusters containing non-Trusted Solaris gateways. This procedure is called *tunneling*. For our purposes, a *cluster* is a contiguous set of either Trusted Solaris hosts and gateways only, or non-Trusted Solaris hosts and gateways only. An edge gateway is a gateway (Trusted Solaris or non-Trusted Solaris) that connects a cluster to a cluster of the other type.

The following figure shows an example of tunneling. The shaded rectangles represent non-Trusted Solaris gateways. The loops with thick lines indicate clusters. Cluster #1 is a non-Trusted Solaris cluster; cluster #2 is a Trusted Solaris cluster.

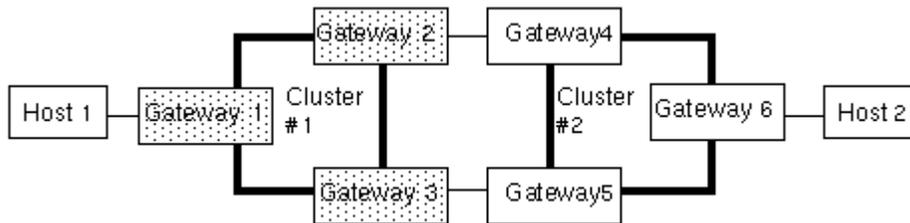


Figure 3-5 Tunneling Example

To transmit data from host #1 to host #2 requires a route through cluster #1, a non-Trusted Solaris cluster, and cluster #2, a Trusted Solaris cluster. This is permitted under these two conditions only:

- All the gateways in the non-Trusted Solaris cluster (in the example, gateways #1, #2, and #3) must have the same security attributes. At start-up, each gateway must have a local file called `/etc/security/tsol/tunnel` containing the addresses of target hosts with which it can connect.
- If there is more than one possible route and the routes enter the non-Trusted Solaris cluster through the same edge gateway and can exit from the cluster through different edge gateways, then the emetrics for these routes must be the same. For example, assume that gateway #4 has an SL range of CONFIDENTIAL to SECRET and gateway #5 has a broader range of ADMIN_LOW to ADMIN_HIGH. Because gateway #1 is a non-Trusted Solaris host, it uses a standard routing table without security attributes and would be unable to distinguish between the route through gateway #4 and the route through gateway #5.

Trusted Solaris Network Commands

The following network commands come from the Solaris environment and have been modified to operate in the Trusted Solaris environment (see the Trusted Solaris Summary section of each man page for the differences).

- `arp(1M)`
- `ifconfig(1M)`
- `ndd(1M)`
- `netstat(1M)`
- `rdate(1M)`
- `route(1M)`
- `snoop(1M)`
- `spray(1M)`

The following network commands are only available in the Trusted Solaris operating environment.

- `tnchkdb(1M)`
- `tnctl(1M)`
- `tnd(1M)`
- `tninfo(1M)`
- `tokmapctl(1M)`
- `tokmapd(1M)`

The `tnd` and `tokmapd` commands launch the trusted network daemon and token mapping daemons, respectively. Token mapping is used when your network is communicating with TSIX host types. The `tnctl` command loads networking information into the kernel caches; the `tninfo` command lets you check this information. The `tnchkdb` examines the network configuration databases for problems. The `tokmapctl` command lets you troubleshoot problems with TSIX token mapping.

Troubleshooting Networks

The Trusted Solaris tools and commands described in this section can help you debug networking problems. For information on the commands, refer to the appropriate man pages. Refer also to Part 3, “Managing Hosts and Networks,” in the *Trusted Solaris Administrator’s Procedures* manual. In addition, standard network debugging commands such as `snoop(1M)`, `ipcs(1)`, and `netstat(1M)` are available in the Trusted Solaris environment.

- To get security information for the source, destination, and gateway hosts in the transmission, use `tninfo(1M)`. You can check whether the information that the kernel is caching is correct. This command is intended to be run at ADMIN_HIGH and effective user ID 0. These restrictions can be overridden by the `file_mac_read`, `sys_trans_label`, and `file_dac_read` privileges. Use `tninfo` as follows:
 - `tninfo -h [<hostname>]` displays the IP Address, port, and template for all hosts or the given host.
 - `tninfo -t <templatename>` displays the following information for all templates or the given template: host type, minimum label (in label and hex format), maximum label (in label and hex format), allowed privileges, and IP label type (RIPSO, CIPSO, or none).
 - `tninfo -k` displays kernel statistics: number of host accreditation check failures, number of network accreditation check failures, and memory allocation statistics.

- To change or check network security information, use the SMC tools to access the `tnrhttp`, `tnrhdb`, and `tnidb` files. If you are not using the NIS+ tables for networking, these changes will take place immediately after you exit from SMC. If you are using NIS+ tables, then the changes will take place when the network daemon next polls the databases or when the system is rebooted. If you wish the change to take place sooner, you can shorten the polling interval using `tnctl(1M)` with the `-p` option on the host that needs the updated information.
- To collect debugging information from the network daemon if the network is already running, use `tnctl(1M)` with the `-d` option. Debugging data is written by default to the file `/var/tsol/tndlog`. Search the log file for failures and other symptoms of problems.
- To check TSIX transmissions, use `tokmapd` with the `-d` option (or `tokmapctl -d`) to create a log and choose an appropriate debugging level. Debugging data is written by default to the file `/var/tsol/tokmapdlog`. Use `snoop(1M)` to check that both source and destination can transmit tokens.

Overview of Trusted NFS Mounting

Mounting filesystems in the Trusted Solaris environment is similar to mounting in the regular Solaris system. You can enter the standard mounting information in the `vfstab` file on the client and the sharing information in the `dfstab` file on the server or you can set up mounting dynamically by using the `mount(1M)` command.

The major differences for setting up mounts in the Trusted Solaris environment are:

- The `vfstab(4)` file is supplemented by a special file called `vfstab_adjunct(4)`, whose purpose is to hold security attributes to be applied to the file system.
- The server needs to have a template assigned in its `tnrhdb` file that it can apply to the client. If you are setting up a mount between two Trusted Solaris hosts, use a template for Trusted Solaris hosts. If you are setting up a mount between a Trusted Solaris host and an unlabeled host, all data is transmitted by default at the single label specified for the unlabeled host in the `tnrhdb` file; however, you can specify different non-label security attributes at mount time using the `vfstab_adjunct(4)` file or the `mount(1M)` command with the `-S` or `-o` option.
- The physical connection between the server and the client must be capable of passing the accreditation checks discussed in “Routing Example” on page 81.
- The `mount(1M)` command requires that UID is 0. Thus you can only run `mount` from a role or user account with an execution profile that includes `mount`, specifies an effective UID of 0, and runs at `ADMIN_LOW`. The `mount` command may need these privileges: `sys_mount`, `file_dac_read`, `file_dac_write`, `file_dac_search`, `file_mac_read`, `file_mac_write`, `file_mac_search`,

`net_privaddr`, `proc_setsl`, `proc_setclr`, and `sys_trans_label`. See `priv_desc(4)` for more information on these privileges. See also “Managing Files and File Systems” in *Trusted Solaris Administrator’s Procedures*

Specifying Security Attributes for Mounting

The `vfstab_adjunct` file and `mount` command with `-S` option let you specify the security attributes for mounts.

The available security attributes are:

- `label`—the label of the files
- `forced privileges`—the set of forced privileges to be applied to executable files in the mounted filesystem
- `allowed privileges`—the set of allowed privileges to be applied to executable files in the mounted filesystem
- `label range`—the range of labels that can be applied to directories and files in the mounted filesystem
- `MLD prefix`—a substitute for `.MLD.` as a prefix for multilevel directories

Administering Auditing

This chapter introduces you to auditing in the Trusted Solaris environment. *Auditing* is the process of capturing user activity and other events on the system, storing this information in a set of files called an *audit trail*, and producing system activity reports to fulfill site security policy. Should a breach of security occur, the audit records may enable you to determine how the breach occurred and which user or users were involved. For a more complete description of the auditing process, refer to the *Trusted Solaris Audit Administration* guide.

- “Planning and Setting Up Auditing ” on page 87
- “Auditing Tools” on page 89

Planning and Setting Up Auditing

Before you set up auditing for your site, you need to:

- Decide which classes of events to audit, including any new classes or events you wish to add to your site.
- Plan where to store the auditing information.
- Define the audit configuration files.

Audit Classes

You need to decide which events you want to audit. You can capture user actions or non-attributable events (that is, events such as interrupts which cannot be attributed to specific users). For the user actions, you can separate successful and failed

transactions. Auditing events are organized into classes in Trusted Solaris. The auditing classes for files fall into these general areas:

- Open for reading
- Open for writing
- Attribute changes
- Creations
- Deletions

You can also create your own classes and events as needed and can rearrange the mapping of classes to events. Other classes keep track of such items as process operations, network events, window operations, IPC operations, administrative actions, logins, logouts, application-defined events, ioctl system calls, program executions, Xserver operations, and miscellaneous events. Because auditing information can take up a lot of disk space, you need to decide carefully which events to audit and select only the classes that contain events necessary for your site's security policy.

Public Objects

One way to reduce the amount of auditing information collected is to specify certain files and directories to be *public objects*. A public object typically contains read-only information, is not modifiable by normal users, and has no implications on security, eliminating the need to track who accesses the object. The system clock is a good example of a public object. When you set the public object flag, any other auditing flags specifying the object are ignored.

Audit Information Storage

The large amount of disk space needed for auditing requires that you plan carefully where the information is going will be collected.

If your site uses individual non-networked workstations, it is recommended that each workstation have a dedicated disk for audit records. The dedicated disk should have at least two partitions:

- a primary storage area
- a partition for holding overflow records

For a network of workstations, you should dedicate at least one separate server for collecting audit information and a second server for administering and analyzing the audit data.

In any case, you should set MAC and DAC protections on the audit files and directories to preserve their integrity and prevent snooping.

Audit Configuration Files

The specifications for auditing at a site are stored in these configuration files, which reside in the `/etc/security` subdirectory:

- `audit_control(4)`—stores audit control information used by the audit daemon, including the preferred order of directories where audit information is stored (the audit daemon uses a directory until the minimum free space warning limit is reached, at which point it stores audit records in the next directory in the list), minimum free space warning limit, system-wide audit flags indicating classes to be audited, and special audit flags for events that cannot be attributed to specific users. The audit flags set in this file are applied to all users. Any exceptions to these flags are set on a per-user basis and specified in the `audit_user` file, which is modified using the User Accounts tool in SMC.
- `audit_user(4)`—stores auditing criteria for users who are exceptions to the auditing specifications in `audit_control`. This information includes user name, events that are always to be audited, and events that are never to be audited.
- `audit_class(4)`—stores audit class definitions, including the class mask (that is, the filter that determines which classes are to be tracked), class name, and description.
- `audit_event(4)`—stores audit event information, including event number, event name, description, and audit flags identifying the audit class.

If you are setting up auditing for a network, there must be identical versions of the `audit_class`, and `audit_event` files on each workstation. Use the SMC to update the `audit_user` site-wide NIS maps and NIS+ tables..

Auditing Tools

This section describes the main utility programs and scripts for administering auditing. Auditing is enabled during system installation. You can enable or disable auditing by editing the `/etc/init.d/audit` script and the `/etc/system` file.

audit and auditd

The `audit(1M)` command is an interface to control the current audit daemon. The audit daemon, `auditd(1M)`, controls the generation and location of audit trail files, using information from the `audit_control` file. The `auditd` command starts the audit daemon (if auditing has been enabled). The `audit` command can halt the daemon, which stops the recording but not the collection of audit records; the `audit` command provides other options as well for controlling the daemon.

The `audit` command lets you

- Reset the first directory in the list of audit storage directories in the `audit_control` file.
- Open a new audit file in the audit directory specified in the `audit_control` file, as last read by the audit daemon.
- Signal the audit daemon to close the audit trail and halt the recording but not the collection of audit records.

auditconfig

The `auditconfig(1M)` command provides a command line interface to get and set kernel audit parameters, including setting various aspects of auditing policy.

audit_startup

The `audit_startup(1M)` script lets you configure auditing parameters during system startup. The script initializes the audit subsystem before the audit daemon is started. This script currently consists of a series of `auditconfig` commands to set the system default policy and download the initial event-to-class mapping. The security administrator can access `audit_startup` by opening the `system_admin` folder in the Application Manager. You can configure it as necessary for your site.

audit_warn

The `audit_warn(1M)` script lets you specify warnings to send out and other actions to take when the audit daemon detects problems. When a problem is encountered, the audit daemon calls `audit_warn` with the appropriate arguments. The option argument specifies the error type. You can specify a list of mail recipients to be notified when an `audit_warn` situation arises by defining a mail alias called `audit_warn` in `aliases(4)`.

praudit

The `praudit(1M)` command prints the contents of an audit trail file in readable form.

auditreduce

The `auditreduce(1M)` command lets you select or merge records from audit trail files from one or more machines. The `merge` function merges audit records from one or more input audit trail files into a single output file. The `select` function lets you select audit records on the basis of criteria relating to the record's content. The `merge` and `select` functions can be combined in a script with the `praudit` command to produce customized reports for your site.

auditstat

The `auditstat(1M)` command displays kernel audit statistics, such as the number of audit records processed and how much memory is being used by the kernel audit module.

Index

A

- accreditation checking
 - networks 83
- accreditation checks
 - overview 79, 81
- accreditation ranges
 - label encodings file 33
 - network interfaces 78
- actions
 - in rights 21
 - administrative
 - accessing 56
 - Add Allocatable Device 54
 - Admin Editor 56
 - Audit Classes 54
 - Audit Control 54
 - Audit Events 54
 - Audit Startup 54
 - Check Encodings 54
 - Check TN Files 54
 - Check TN NIS+ Tables 54
 - Configure Selection Confirmation 54
 - Create NIS+ Client 54
 - Create NIS+ Server 54
 - Edit Encodings 55
 - Name Service Switch 55
 - Set Mail Options 55
 - Set Mount Attributes 55
 - Set Mount Points 55
 - Set Tsol Gateways 55
 - Share Filesystems 55
 - running 18
 - Add Allocatable Device action
 - described 54
 - add_allocatable(1M) command 54
 - Admin Editor action 56
 - described 54
 - ADMIN_HIGH label
 - defined 33
 - ADMIN_LOW label
 - defined 33
 - administration labels
 - defined 33
 - administrative actions
 - accessing 56
 - administrative roles
 - defined 17
 - adminvi(1M) command
 - described 56
 - adminvi(1M) command
 - default editor for administrative actions 56
 - adornments
 - defined 37
 - Application Manager
 - accessing applications 18
 - applications
 - accessing 18
 - administering 21
 - assigning privileges
 - overview 29
 - audit classes
 - overview 88
 - Audit Classes action
 - described 54
 - audit command

- overview 90
- Audit Control action
 - described 54
- Audit Events action
 - described 54
- Audit Startup action
 - described 54
- audit_class file
 - overview 89
- audit_class(4) file
 - action for editing 54
- audit_control(4) file
 - action for editing 54
- audit_event file
 - overview 89
- audit_event(4) file
 - action for editing 54
- audit_startup script
 - overview 90
- audit_startup(1M) command
 - action for editing 54
- audit_user file
 - overview 89
- audit_warn script
 - overview 90
- auditconfig command
 - overview 90
- auditing
 - configuration files 89
 - overview 87, 91
 - planning 87, 89
 - process privileges 30
 - system privileges 31
 - tools 89, 91
- auditreduce command
 - overview 91
- auditstat command
 - overview 91
- authorizations
 - in software administration 21
 - in rights 21
 - categories 7, 26, 27
 - defined 16
 - overview 25, 29

B

booting the workstation

- system privileges 31
- broadcast messages
 - network privileges 30

C

- Check Encodings action
 - described 54
- Check TN Files action
 - described 54
- Check TN NIS+ Tables action
 - described 54
- chk_encodings(1M) command
 - action for invoking 54
- CIPSO
 - host type 74
- classification label component
 - defined 32
- clearances
 - label overview 31
 - network interfaces 78
 - remote host templates 75
- closed networks
 - defined 72
- colormaps
 - window privileges 31
- commands
 - in rights 21
- compartment label component
 - defined 32
- component definitions
 - label encodings file 33
- configuration management
 - system privileges 31
- Configure Selection Confirmation action
 - described 54
- console redirection
 - system privileges 31
- covert channel delays
 - process privileges 30
- Create NIS+ Client action
 - described 54
- Create NIS+ Server action
 - described 54
- customizations
 - label encodings file 33

D

DAC

- defined 16

databases

- tnidb database 78
- tnrhdb(4TSOL) 77

defaultrouter file

- action for editing 55

defaults

- privileges 30
- rights 22

Device Allocation Manager

- overview 45

device_allocate(4) file

- action for editing 54

device_maps(4) file

- action for editing 54

devices

- allocation 40
- clean scripts 44, 45
- configuration files 40
- label ranges 40
- overview 40

dfstab(4) file

- action for editing 55

DGA (direct graphics access)

- window privileges 31

dominance of labels

- overview 32

dtpad(1) command

- using in administrative actions 56

E

Edit Encodings action

- described 55

emetrics

- defined 79

/etc/defaultrouter file

- action for editing 55

/etc/mail/sendmail.cf file

- action for editing 55

/etc/motd file

- action for editing 55

F

File Manager

- accessing applications 18

file systems

- privileges 30

font paths

- window privileges 31

Front Panel

- accessing applications 18

G

gateway host

- defined 72

GIDs

- in rights 21
- network interfaces 78

H

host types

- cipso 74
- networking 73
- ripso 74
- sun_tsol 74
- tsix 74
- unlabeled 74

I

inter-window movement

- window privileges 31

IP addresses

- tnrhdb database 77

IP labels

- remote host templates 75

IPC

- privileges 30

L

label encodings file

- overview 33

label ranges

- assigning to devices 40

- defined 34

- in rights 21

label_encodings(4) file

- action for editing and checking 55

labels

- classification component 32
- compartment component 32
- dominance 32
- overview 31
- privileges for overriding 30
- relationships 32
- remote host templates 75
- well-formed 34
- linking
 - system privileges 31
- loadable modules
 - system privileges 31
- login
 - authorizations 26

M

MAC

- defined 16

maximum labels

- remote host templates 75

media labeling

- clean scripts 45

message queues

- overriding restrictions 30
- system privileges 31

minimum labels

- remote host templates 75

motd file

- action for editing 55

mounting

- defined for Trusted Solaris 78
- overview 85, 87

multilevel port bindings

- network privileges 30

N

Name Service Switch action

- described 55

ne 81

network interfaces

- defined 72
- tnidb(4TSOL) 78

networks

- example 81, 83
- modified Solaris commands 83
- overview 71, 85

- privileges 30
 - Trusted Solaris commands 84, 85
- niscat(1) command
 - action to invoke 55
- nisclient(1M) command
 - action for creating NIS+ client 54
- nispopulate(1M) file
 - action for invoking 55
- nissserver(1M) file
 - action for invoking 54
- nsswitch.conf(4) file
 - action for editing 55

O

object-reuse

- clean scripts 45

open networks

- defined 72

P

packets

- standard Solaris 73
- Trusted Solaris 73

pattr command

- overview 69

permissions

- overriding 30

plabel command

- overview 69

Populate NIS+ Tables action

- described 55

port bindings

- network privileges 30

ppriv command

- overview 69

pprivtest command

- overview 69

praudit command

- overview 90

primary administrator

- defined 20

printing definitions

- label encodings file 33

privilege sets

- defined 29

- privileges
 - categories 30
 - file system privileges 30
 - network privileges 30
 - process privileges 30
 - system privileges 31
 - System V IPC privileges 30
 - window privileges 31
 - defined 16
 - inheritable 30
 - network interfaces 78
 - overview 29
 - in rights 21
 - remote host templates 75
- processes
 - privileges 30
- profile shell
 - defined 18
- profiles
 - overview 21

R

- remote hosts
 - tnrhdb(4TSOL) 77
- resolv.conf(4) file
 - action for editing 55
- rights
 - All
 - described 22
 - All Authorizations
 - described 22
 - Audit Control
 - described 22
 - Audit Review
 - described 22
 - Basic Actions
 - described 22
 - Basic Commands
 - described 23
 - Convenient Authorizations
 - described 23
 - Cron Management
 - described 23
 - Custom Admin Role
 - described 23
 - Custom Admin Secadmin Role
 - described 23

- Custom Oper Role
 - described 23
- Custom Root Role
 - described 23
- default 22
- Device Security
 - described 23
- Enable Login
 - described 23
- File System Management
 - described 23
- File System Security
 - described 23
- Mail Management
 - described 23
- Maintenance and Repair
 - described 23
- Media Backup
 - described 23
- Media Restore
 - described 23
- Network Management
 - described 24
- Object Access Management
 - described 24
- Object Label Management
 - described 24
- Object Privilege Management
 - described 24
- Outside Accred
 - described 24
- overview 19, 21
- Privileged Shells
 - described 24
- Process Management
 - described 24
- User Management
 - described 24
- User Security
 - described 24
- Rights Manager
 - assigning inheritable privileges 30
- RIPSO
 - host type 74
- roles
 - overview 20
- route(1M) command

- 55
- route(1M)TSOL
 - example 81
- routing
 - example 81
 - loading data at boot time 78
 - overview 78, 83
 - tables 79
 - through non-Trusted Solaris clusters 82
- routing commands
 - examples 81

S

- SAMP (Security Attribute Modulation Protocol)
 - Trusted Solaris data packets 74
- security administrator
 - defined 20
- security attributes
 - in data packets 74
- security domain
 - defined 72
- security policy
 - overriding 30
- sel_config(4) file
 - action for editing 54
- semaphore sets
 - overriding restrictions 30
- sendmail(1M) command 55
- sendmail.cf file
 - action for editing 55
- session range
 - defined 34
- Set Daily Message action
 - described 55
- Set Default Routes action
 - described 55
- Set DNS Server action
 - described 55
- Set Mail Options action
 - described 55
- Set Mount Attributes action
 - described 55
- Set Mount Points action
 - described 55
- Set Tsol Gateways action
 - described 55

- Share Filesystems action
 - described 55
- shared memory regions
 - overriding restrictions 30
- SLs
 - defined 16
 - label overview 31
 - network interfaces 78
- SMC Computers and Networks Tool
 - network configuration databases 77
- SMC Interface Manager Tool
 - network configuration databases 77
- sun_tsol host type
 - defined 74
- system administrator
 - defined 20
- system operator
 - defined 20
- system security
 - privileges 31
- system security configuration
 - networks 72
- System V IPC
 - privileges 30

T

- /tmp directory
 - as an MLD 35
- tnchkdb(1M) command
 - action for checking NIS+ tn* tables 54
 - action for checking local tn* files 54
- tnidb database 78
- tnidb(4) file
 - action for checking local version 54
- tnrhdb database 77
- tnrhdb(4) file
 - action for checking local version 54
 - action for checking NIS+ version 54
- tnrhtp(4) file
 - action for checking NIS+ version 54
 - action for checking local version 54
- trusted applications
 - defined 21
- trusted path attribute
 - defined 17
- trusted_edit shell script

use in administrative actions 56
tsix host type
 defined 74
tsolgateways(4) file
 action for editing 55
tunneling
 described 82

U

UIDs
 network interfaces 78
 in rights 21
unlabeled host type
 defined 74
users
 session range 34

V

vfstab_adjunct(4) file

 action for editing 55
View Table Attributes action
 described 55
View Table Contents action
 described 55

W

well-formed labels
 defined 34
windows
 privileges 31
workstation configuration
 system privileges 31

X

X server
 window privileges 31