



Trusted Solaris Administrator's Procedures

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

Part No: 816-1048-10
November 2001

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road Palo Alto, CA 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, Solaris Management Console 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. PostScript(TM) is a trademark or registered trademark of Adobe Systems, Incorporated, which may be registered in certain jurisdictions.

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.

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, Solaris Management Console 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. PostScript est une marque de fabrique d'Adobe Systems, Incorporated, laquelle pourrait être déposée dans certaines juridictions. in the United States and other countries.

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.



010928@2471



Contents

Preface 17

1 Administering in a Role 23

Administering Systems in an Administrative Role 23

Accessing Administration Tools 24

Administering Remote Systems 24

Administering as a Role (Tasks) 25

▼ To Log In and Assume a Role 25

▼ To Leave an Administrative Role 29

▼ To Launch the Solaris Management Console 30

▼ To Launch Local Administrative Actions 32

▼ To Edit a Local File 33

▼ To Work at a Different Label 33

▼ To Enable Any Role to Log In Remotely 35

▼ To Log In Remotely From the Command Line 35

▼ To Launch Administrative Actions Remotely 36

2 Administering Security Requirements 39

Enforcing Security Requirements 39

Training Users About Security Requirements 40

Using Email 40

Enforcing Password Requirements 41

Protecting Information 41

Protecting Passwords 42

Administering Groups 42

Deleting Users	43
Changing Number of Allowable Password Tries	43
Managing the Relabeling of Files	45
sel_config File Sections	47
Extending Authorizations and Privileges	48
Adding New Authorizations	48
Adding New Privileges	49
Changing CDE Defaults	51
Customizing the Workspace Menu	51
Customizing the Front Panel	52
Changing and Accessing Security Information (Tasks)	53
▼ To Change the Allowed Number of Password Tries	53
▼ To Prevent Account Locking for Individuals	53
▼ To Prevent Account Locking for All User Accounts	54
▼ SPARC: To Enable Keyboard Shutdown	54
▼ To Prevent Logins From Being Disabled After a Reboot	55
▼ To Change Configurable Kernel Switch Settings	56
▼ To Modify the Selection Configuration File	57
▼ To Add an Authorization to the Environment	57
▼ To Add a Privilege to the Environment	59
▼ To Customize the Workspace Menu	60
▼ To Get a Hexadecimal Equivalent for a Label	61
▼ To List a User's Home Directory SLDs and Their Labels	61
3 Managing User Accounts	63
Setup Before Creating User Accounts	63
Decisions to Implement Before Creating Users	64
Decisions to Implement Before Users Log In	64
Managing Default User Security Attributes	65
Label Encodings File Defaults	65
policy.conf File Defaults	66
Managing Remote Logins	67
Managing Initialization Files	67
Controlling the Sourcing of Startup Files	69
.dtpfile Files	69
The Sourcing of Startup Files for the Profile Shell User	70
Controlling Which Startup Files Are Read When a Shell Comes Up	70

Forcing dtterm to Source \$HOME/.login or .profile	71
Administering Skeleton Directories	71
Accessing All Man Pages	72
Using .copy_files and .link_files	72
Administering cron, at, and batch Jobs	73
Running a Job with a Profile Shell	74
Running Privileged Commands in Scheduled Jobs	74
How the UNIX Domain Socket is Used for Communications	75
Permitting Users to Access Others' Jobs	75
Conditions for Access to Other's Jobs	75
Assigning the SMC to Normal User Accounts	76
Preparing for User Accounts (Tasks)	77
▼ To Modify Default User Label Attributes	77
▼ To Modify policy.conf Defaults	77
▼ To Set Up Startup Files for Users	77
▼ To Invoke .login or .profile During Login	79
▼ To Force dtterm to Launch New Shells as Login Shells	80
▼ To Customize Shell Initialization Files for Users	80
▼ To Enable a User to Track Others' Jobs on a System	81
▼ To Enable a User to Track All Others' Jobs	81
4 Managing Users and Rights With SMC	83
Before Setting Up User Accounts	83
Adding or Modifying a User Account	84
Assigning Passwords to Users	86
Assigning Rights to Users	86
Assigning Roles to Users	87
Assigning Trusted Solaris Attributes to Users	87
Assigning Audit Classes to Users	87
Adding or Modifying a Rights Profile	87
Managing Users and Rights (Tasks)	88
▼ To List All Rights	88
▼ To Create a Help File for a Rights Profile	89
▼ To Create a Rights Profile	89
▼ To Modify a Rights Profile	90
▼ To Create a User Template	91
▼ To Add a User Account	92

▼ To Modify a User Account	92
▼ To Assign a Right to a User	93
▼ To Assign an Authorization to a User	93
5 Managing Roles	95
Roles and the Trusted Path Attribute	95
Allowing Remote Logins by Administrative Roles	96
Creating a New Role	96
Modifying a Role With the SMC	97
Customizing Profiles for the Recommended Roles	97
Enabling Role Assumption from Untrusted Systems	98
Managing Roles (Tasks)	98
▼ To Alias vi to adminvi	98
▼ To Assign the trusted_edit Editor to a Role	99
▼ To Alias vi to trusted_edit	100
▼ To List All Roles	100
▼ To Modify a Role	101
▼ To Configure a New Role	102
▼ To Enable a Role to Administer NIS+	102
▼ To Enable Remote Role Assumption from Untrusted Systems	103
6 Managing Mail	105
Managing Trusted Solaris Mail Features	105
.mailrc Is at User's Minimum Label Only	106
The Solaris Management Console Manages Mail Aliases	106
Users Cannot Read Email Below Minimum Label	106
Users Cannot List the Mail Queue	107
dtmail is the Default Mail Application	107
Troubleshooting Mail Problems	108
Tracing Mail Delivery Difficulties	108
Tracing sendmail's Activities	108
Debugging sendmail	110
Managing Mail (Tasks)	110
▼ To Enable the IMAP Server to Authenticate Users	110
▼ To Configure Users To Receive Mail Below Their Minimum Labels	110
▼ To Modify a Mail Alias	111

▼ To Permit Users to See the Mail Queue	111
▼ To Troubleshoot Mail Delivery Difficulties	112
▼ To Trace sendmail for Trusted Solaris Information	112
▼ To Check Network Connections for Sending Mail	113
▼ To Troubleshoot Loss of Mail Icons	116
▼ To Create a Multilevel Action for the Alternate Mail Application	116
▼ To Substitute an Alternate Mail Application for All Users	118
▼ To Install an Alternate Mailer in the Front Panel	119
7 Managing Computers and Networks	121
Managing Trusted Network Communications	121
SMC Tools for Administering Computers and Networks	122
Meeting the Goals of Trusted Networking	123
Understanding Security Attributes Assigned to Computers	124
Host Types	125
Computer Accreditation Range	126
Domain of Interpretation (DOI)	126
Default Label	127
Default Clearance	127
Forced Privileges	127
Allowed Privileges	128
Advanced Security Attributes	128
Using IP Labels in Trusted Routing	129
Default Templates	130
Default Templates for Trusted Solaris Systems	130
Default Templates for Unlabeled or RIPS0 Computers	130
Wildcard Entry and Prefix Length	131
CIPSO Labels in Packets	132
RIPS0 Labels in Packets	133
Understanding Security Attributes Assigned to Network Interfaces	134
Network Interface Accreditation Range	134
Default Security Attributes	135
Accreditation Checks	136
MAC Enforcement on Outgoing Messages	137
MAC Checks on Messages Being Forwarded	137
MAC Enforcement on Incoming Messages	138
Administering Routing	139

Background on Routing	139
Choosing Routers	140
Enabling a Single-Label Gateway to Forward Packets at Multiple Labels	143
8 Specifying Routing and Security for Remote Computers	145
Assigning Security Attributes to Remote Hosts and Network Gateways	145
Setting Up Templates	146
Storing Network Information	147
Modifying the Boot-Time Tnrhdb File	147
Setting Up Tunneling	148
Managing Trusted Networking (Tasks)	148
▼ To Open the Security Families Tool	148
▼ To Construct Templates for Hosts	149
▼ To Assign Templates to Hosts	149
▼ To Create a Wildcard Entry for Remote Hosts	150
▼ To Change the tnd Polling Interval	150
▼ To Replace the 0.0.0.0 Entry in the Local Tnrhdb File	151
▼ To Configure a Network Interface	154
▼ To Set Up Static Routes with Emetrics	155
▼ To Set Up Tunneling	156
9 Managing Files and File Systems	159
Requirements Unique to Trusted Solaris File Systems	159
Specifying Security Attributes on Files and File Systems	160
Security Attributes on Files and Directories	161
Specifying Security Attributes on Files and Directories	161
Security Attributes on File Systems	163
Mounting File Systems in the Trusted Solaris Environment	167
Mount Options Used for Protection	168
Summary of Attributes on Various File System Types	169
Trusted Solaris Attribute Precedence Rules	171
Trusted Solaris Software and NFS	172
Sharing Directories	173
Troubleshooting Mount Failures	173
Managing Files and File Systems (Tasks)	174
▼ To Back Up Files	174

▼ To Restore Files	174
▼ To Change Labels and Privileges With the File Manager	175
▼ To Set Security Attributes While Creating a Local File System	176
▼ To Set Security Attributes on a File System	177
▼ To Specify Mount-time Security Attributes on the Command Line	178
▼ To Specify Mount-time Security Attributes in the <code>vfstab_adjunct</code> File	178
▼ To Share a Directory	179
▼ To Mount a TMPFS File System Using the Command Line	180
▼ To Mount a CD-ROM with a HSFS File System	180
▼ To Automatically Launch a CD Player for an Audio CD-ROM	180
▼ To Listen to an Audio CD as any User or Role	181
▼ To Troubleshoot Mount Failures	181
10 Managing Name Services	183
Managing Multiple Trusted Solaris Computers in a Security Domain	183
Managing Standalone Trusted Solaris Computers	184
Enabling the root Role or a New Role to Administer a Name Server	184
Trusted Solaris NIS Maps and NIS+ Tables	185
Managing Name Services (Tasks)	185
▼ To Enable Domain Administration from a Client	185
▼ To Save and Restore NIS Maps	186
▼ To Save and Restore NIS+ Tables	186
▼ To Use NIS and NIS+ Administrative Actions	188
11 Managing Printing	189
Requirements Unique to Trusted Solaris Printers	189
Configuring Printers in a Trusted Solaris Environment	190
Allowing the Printing of PostScript Files	191
Adding Support for Additional File Types	191
Setting Up Printers That do not Support Security Features	191
Managing Network Printers	192
Controlling Whether Security Information is Printed on Print Jobs	192
Print Job Information on Banner and Trailer Pages	193
Permitting Safe Jobs to Be Printed Without Labeled Pages	195
Managing Printing (Tasks)	196
▼ To Set Up Printing to a Non-Trusted Solaris Server	196

▼ To Launch the Printer Administrator Action	196
▼ To Configure an Attached Printer	196
▼ To Configure a Network Printer for Labeled Output	197
▼ To Configure a Restricted Label Range for a Printer	198
▼ To Add Access to a Remote Printer	200
▼ To Enable Some Users to Print Without Banners and Trailer Pages	200
▼ To Assign Printing-Related Authorization(s) to an Account	201
▼ To Suppress the Printing of Page Labels on All Print Jobs	201
▼ To Allow Some Users to Print Jobs Without Page Labels	201
▼ To Set Up Public Print Jobs from an Unlabeled Print Server	202
 12 Managing Devices	203
Controlling Access to Devices	203
Setting a Label Range	204
Managing Device Access Policies	204
Initial Device Configuration Decisions	205
Managing Devices	206
Making a Device Available	206
Using the Device Allocation Manager	207
Handling of Allocated Devices at Boot	209
Authorizing Device Allocation	210
Enforcing Device Security	211
Recovering From the Allocate Error State	211
Using Device-Clean Scripts	211
Device-Clean Script for Tape Devices	212
Device-Clean Scripts for Floppy Disks and CD-ROM	212
Device-Clean Script for Audio	213
Writing New Device-Clean Scripts	213
Mounting an Allocated CD-ROM Device	214
Mounting an Allocated Floppy Device	214
Device-related Commands, Databases, and Files	214
Ancillary Files for Allocatable Devices	215
Managing Devices (Tasks)	216
▼ To Save Files With Security Attributes to a Tape	216
▼ To Set or Modify Device Policy for a Device	217
▼ To Revoke or Reclaim a Device	218
▼ To Play an Audio CD	218

▼ To Add a Device	218
▼ To Add Site-Specific Authorizations to a Device	220
▼ To Configure a Serial Line for Logins	220
▼ To Assign Device Authorizations to an Account	222
▼ To Prevent File Manager Display After Device Allocation	223
▼ To Change or Add a Device Clean Script	224
13 Adding Software	225
Types of Software	225
Administrator Role Responsibilities	226
Security Administrator Role Responsibilities	226
Privilege Enabling Mechanisms	227
System Shell	227
Profile Shells	228
Trusted Processes in the Window System	228
Trusted Libraries	229
Assigning Privileges	229
Giving Forced Privileges to an Executable File	229
Assigning Inheritable Privileges to a Command or Action	230
Passing Privileges to Child Processes	230
Passing Privileges to Another Program	231
Not Passing Forced Privileges via Shell Scripts	231
Creating and Using Shell Scripts	232
Summary of Shell Script Behavior in the Trusted Solaris Environment	233
Using Profile Shell Scripts	234
Editing Executables With Inheritable Privileges	235
Testing New Software for Security	235
Evaluating a Program for Security	236
Considering When to Add Privilege	237
Running a Program As Root	238
Cooperating to Create a Trusted Program	238
Adding Trusted Actions	239
Finding Which Privileges a Program Needs	241
Making Libraries Trusted	241
Adding Boot Commands	242
Adding Commands to the <code>inittab</code> File	243
Adding Commands to <code>/etc/init.d</code> Scripts	243

Adding Services to the <code>inet</code> Daemon	244
Managing Software (Tasks)	245
▼ To Mount a CD-ROM for Adding a Package	245
▼ To Give Forced Privileges to a Command	246
▼ To Create a New File Edit Action	246
▼ To Add Actions Outside of the <code>System_Admin</code> Folder	248
▼ To Make New Actions Available to the Rights Tool	249
▼ To Write a Profile Shell Script	249
▼ To Write a Standard Shell Script that Runs Privileged Commands	251
▼ To Save and Restore Privileges When Editing a File	252
▼ To Find Out Which Privileges a Program Needs	252
▼ To Make a Library Directory Trusted	255
▼ To Add Commands to the <code>/etc/inittab</code> File	256
▼ To Run <code>rc</code> Scripts With Security Attributes	257
▼ To Add Services to the <code>inetd.conf</code> File	258
▼ To Install a Java Jar File	258
 Index	 259

Tables

TABLE 2-1	Conditions for Moving Files Between File Managers	46
TABLE 2-2	Conditions for Moving Selections Between Windows	47
TABLE 2-3	Configurable Trusted Solaris Switches in /etc/system	56
TABLE 3-1	Security Defaults for Users in the label_encodings File	65
TABLE 3-2	Security Defaults for Users and Roles in the policy.conf	66
TABLE 3-3	Startup Files Read at Shell Initialization	70
TABLE 3-4	Planning Worksheet for .copy_files and .link_files	73
TABLE 4-1	Configurable User Attributes in a Wizard versus a Template	85
TABLE 4-2	User Security Attributes Assigned after Creation	85
TABLE 7-1	Host Types, Protocols, and Notes	125
TABLE 7-2	Wildcard Address, Netmask, and Prefix Length	131
TABLE 7-3	Protection Authority Flags	133
TABLE 9-1	Trusted Solaris File and Directory Attributes	161
TABLE 9-2	Variable File System Security Attributes with Defined Settings	164
TABLE 9-3	Attributes Assignable to Fixed File Systems	166
TABLE 9-4	Mount Types, Examples, and Notes	167
TABLE 9-5	Mount Restrictions, Default Values	168
TABLE 9-6	Attributes Supported by the Supported File System Types	170
TABLE 9-7	KEY to the File System Attributes Table	170
TABLE 11-1	Tasks for Configuring Printers	190
TABLE 11-2	Modifiable Printing Features	194
TABLE 12-1	Default Device Access Policy	205
TABLE 12-2	Requiring Separate Authorizations for Local and Remote Device Use	210
TABLE 12-3	Specifying Only Local Allocation of the Audio Device	210
TABLE 12-4	Device-related Commands and Databases	215

TABLE 12-5	Required Ancillary File Characteristics for Devices	215
TABLE 12-6	Default Profiles that Include Device Allocation Authorization	222
TABLE 12-7	Default Profiles for Administering Devices	223
TABLE 12-8	Default Profiles for Creating Devices	223
TABLE 13-1	Constraints on Actions in the Trusted Solaris Environment	240

Figures

FIGURE 1-1	Workstation Information Dialog Box	26
FIGURE 2-1	File Manager Selection Confirmer	45
FIGURE 3-1	How \$HOME/.dtpfile is Installed	69
FIGURE 3-2	How \$HOME/.dtpfile is Bypassed for Profile Shell Users	70
FIGURE 6-1	Sendmail Data Flow Example	109
FIGURE 7-1	Solaris Management Console Tools	122
FIGURE 7-2	How a Host Determines Which Type of Routing to Do	142
FIGURE 8-1	Interface Manager with Default Security Attributes	155
FIGURE 9-1	File Manager Selected Menu for an Authorized User	162
FIGURE 9-2	Trusted Solaris Attribute Precedence Rules	172
FIGURE 11-1	Job's Label Printed on Body Pages	193
FIGURE 11-2	Typical Print Job Banner Page	193
FIGURE 11-3	Differences on a Trailer Page	194
FIGURE 12-1	Device Allocation Configuration Dialog	207
FIGURE 12-2	Solaris Management Console Tools	220
FIGURE 13-1	How an Unprivileged Program Can Pass On Privileges	231
FIGURE 13-2	How Forced Privilege Shell Scripts Are Prevented from Passing On Privileges	231
FIGURE 13-3	How Shell Scripts Pass Inheritable Privileges Using a Profile Shell	233

Preface

This *Trusted Solaris Administrator's Procedures* guide provides procedures for managing users and hosts while maintaining the security of information within the Trusted Solaris™ environment.

Who Should Use This Book

This book is used by administrators who are able to assume any of the Trusted Solaris administrative roles. This book describes how to do the unique Trusted Solaris administrative tasks that are an essential part of protecting the security of the system.

Before You Read This Book

- **Understand Solaris 8 administration, CDE, Solaris Management Console™, and NIS+.**

The procedures in this guide are unique to Trusted Solaris administration, and often add to Solaris procedures. An administrator should already understand how to administer the Solaris operating environment, how to use and administer the Common Desktop Environment (CDE) window system, how to use Solaris Management Console administration tools, and administer a name service.

Note – AnswerBooks for the above-mentioned products that are bundled into Trusted Solaris are available on the *Trusted Solaris 8 4/01 AnswerBook* CD, which is shipped with the Trusted Solaris 8 4/01 product CD.

- **Read and understand the basic concepts and procedures for using the system, as described in the *Trusted Solaris User's Guide*.**

Administrators should know how to work in the Trusted Solaris environment as a normal user.

- **Read and understand the administrative concepts described in the *Trusted Solaris Administration Overview*.**
- **Understand how administrative tasks are divided among roles at your site.**

Each procedure identifies which role is assigned to the task in the default configuration. The Security Administrator role is responsible for informing administrators if the default administrative roles have been reconfigured.

How This Book Is Organized

Chapter 1 reviews how to work in an administrative role.

Chapter 2 describes general security mechanisms and common procedures to harden the system.

Chapter 3 describes how to prepare for setting up user and role accounts, and how to administer startup files and batch jobs.

Chapter 4 describes how to manage rights, roles, and users using the Solaris Management Console.

Chapter 5 describes how to modify and extend the powers of a role.

Chapter 6 describes the differences between standard Solaris and Trusted Solaris mail administration.

Chapter 7 reviews concepts that apply to managing communications and shows how trusted communications are configured between a Trusted Solaris system and multiple networks.

Chapter 8 describes how to specify the security attributes for hosts and how to set up routing for trusted network communications.

Chapter 9 describes the extended file system security attributes, how to set up mounts, and how to specify extended security attributes.

Chapter 10 describes how NIS and NIS+ name services can be used to centrally administer a Trusted Solaris network.

Chapter 11 describes how to configure printing for labeled and unlabeled jobs.

Chapter 12 describes how to manage devices, and includes how to set the label range on printers and computers.

Chapter 13 describes how to assess software for trustworthiness. Software includes Sun software products, other UNIX[®] applications, new trusted programs, CDE actions, and shell scripts. The chapter also describes the Trusted Solaris privilege mechanism.

Related Books

- *Trusted Solaris User's Guide*

The rest of the Trusted Solaris administrator's document set:

- *Trusted Solaris Administration Overview*
- *Trusted Solaris Audit Administration*
- *Trusted Solaris Developer's Guide*
- *Trusted Solaris Installation and Configuration*
- *Trusted Solaris 8 4/01 Installation and Configuration on the Sun Enterprise 10000*
- *Trusted Solaris Label Administration*
- *Trusted Solaris 8 4/01 Reference Manual*
- *Trusted Solaris 8 4/01 Release Notes*
- *Trusted Solaris 8 4/01 Transition Guide*
- *Compartmented Mode Workstation Labeling: Encodings Format*

Ordering Sun Documents

Fatbrain.com stocks documentation from Sun Microsystems, Inc.

For a list of available documents and how to order them, visit
<http://www1.fatbrain.com/documentation/sun>.

Accessing Sun Documentation Online

The docs.sun.comSM Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is <http://docs.sun.com>.

Type Styles Used in Text and Examples

The following table shows and explains the type styles used in this guide.

TABLE P-1 Typographic Conventions

Type Face	Meaning	Example
Literal	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>hostname%</code> You have mail.
UserType	What you type, contrasted with on-screen computer output	<code>hostname% su</code> Password:
Variable	Argument name in a command-line. You replace the argument with a real name or value.	To delete a file, enter <code>rm filename</code> . <code>hostname% rm myfile</code>
Title or Emphasis	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.

Trusted Solaris Prompts

The following table shows the Trusted Solaris prompts.

Shell	Prompt
C shell prompt	<i>hostname%</i>
Bourne shell and Korn shell prompt	\$
Profile shell prompt	\$
root prompt	#

Administering in a Role

This chapter describes how to assume an administrative role and use administrative tools. It also describes what to consider when logging in remotely to assume a role. This chapter contains the following procedures:

- “To Log In and Assume a Role” on page 25
- “To Leave an Administrative Role” on page 29
- “To Launch the Solaris Management Console” on page 30
- “To Launch Local Administrative Actions” on page 32
- “To Edit a Local File” on page 33
- “To Work at a Different Label” on page 33
- “To Enable Any Role to Log In Remotely” on page 35
- “To Log In Remotely From the Command Line” on page 35
- “To Launch Administrative Actions Remotely” on page 36

Administering Systems in an Administrative Role

As described in the *Trusted Solaris Administration Overview*, users administer Trusted Solaris systems after having assumed a role. The programs and tools available to a role have a special property, the *trusted path attribute* to enable the commands to succeed. In the Trusted Solaris environment, the role root has very limited powers. The Security Administrator (usually called secadmin) and the System administrator (usually called admin), roles perform most tasks.

A user who can assume a role chooses the Assume *Rolename* Role option from the Trusted Path (TP) menu in the Front Panel, and types a password for the role. When the password is correct, an administrative role workspace at the label ADMIN_LOW becomes active with the trusted path attribute. The shell available in the role

workspace is called a *profile shell*, which enables commands to execute securely. Each role can use only those tools in the rights profile(s) that are assigned to that role.

Accessing Administration Tools

The following table lists the Trusted Solaris administrative tools and where their use is described.

Solaris Management Console tool or equivalent commands, such as <code>smuser(1M)</code> and <code>smrole(1M)</code> .	Used for most configuration of user accounts, hosts, and networks. Can update local files or name service databases. Can also launch legacy applications: <code>dtterm(1)</code> and <code>dtappsession(1)</code> .	<i>Note:</i> Authorizations are used to control which tools or fields can be accessed by each role in the Solaris Management Console and which options can be used in the equivalent commands. See “To Launch the Solaris Management Console” on page 30.
Trusted Solaris administrative actions in the System_Admin Folder in the Application Management folder	Used to edit local files that the Solaris Management Console does not manage, such as <code>/etc/system</code> .	See “To Launch Local Administrative Actions” on page 32.
Administrative commands and actions.	Used to perform tasks not covered by the Solaris Management Console or System_Admin programs.	See <i>man pages</i> section 1M: <i>System Administration Commands</i> .

Administering Remote Systems

Administrators can administer from remote hosts in several ways that are described in this guide, as summarized below:

- After logging in to the local host and assuming a role, administrators can log in to a remote host from a terminal in their role workspace and use the commands `rlogin(1)`, `telnet(1)`, or `ftp(1)`. See “To Log In Remotely From the Command Line” on page 35 for how roles can log in remotely and work on the command line.
- From a local host’s CDE login screen, anyone can log directly into the CDE window system on a remote host (as described in “To Log In and Assume a Role” on page 25). . This works just as it does on a Solaris system.

After CDE remote login is complete, the CDE window environment from the remote host displays on the screen of the local host. An administrator can then assume a role from the Trusted Path menu and work as if logged in directly to the remote host.

- Administrators can launch a Solaris Management Console (SMC), server that is running on a remote host. Accessing the SMC is described in “To Launch the Solaris Management Console” on page 30.

The Application Manager can be started remotely by double-clicking the Application Manager icon from the Legacy Application list in the SMC. The Application Manager contains the System_Admin folder, a collection of programs to modify local system files that are not managed directly by the Solaris Management Console.

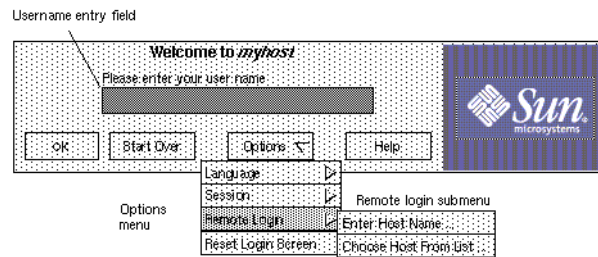
- While working in a role workspace on a local host, the role can use the `dtappsession(1)` command to launch an Application Manager that runs and makes changes on the remote host. The `dtappsession` script starts an independent instance of the CDE Application Manager that runs on the remote host and displays on the local host. Unlike CDE remote login, `dtappsession` enables the administrator to work remotely within a local login session.

`dtappsession` is useful when a remote host does not have a monitor. For example, `dtappsession` is often used instead of CDE remote login when administering domains on large servers, such as a Sun Enterprise™ 10000.

Administering as a Role (Tasks)

▼ To Log In and Assume a Role

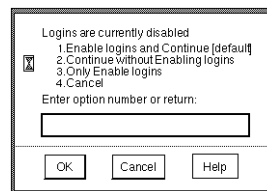
1. (Optional) If you are starting a login session on a remote host, choose **Remote Login** from the **Options** drop-down menu. You can either select your host name from a list (by choosing **Choose Host From List**) or choose **Enter Host Name** and type it yourself.



2. Type your username in the field, then supply a password when prompted.

Note – If a message appears stating that logins are currently disabled, you are not currently authorized to enable logins. Ask the security administrator to give you the needed authorization, or ask an authorized person to enable logins.

3. Choose one of the options in the Enable Options dialog box shown in the following figure, then click OK.



4. Review the information in the Workstation Information dialog box shown in the following figure.

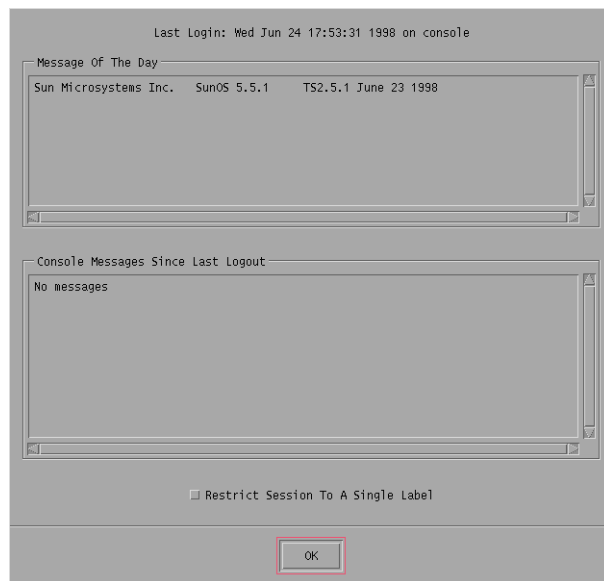


FIGURE 1-1 Workstation Information Dialog Box

Investigate any suspicious logins, messages that could indicate inappropriate activities, and the date and time of the last login to see if it occurred at an unusual time of day, for example. Check the message of the day and the console messages since the last logout.

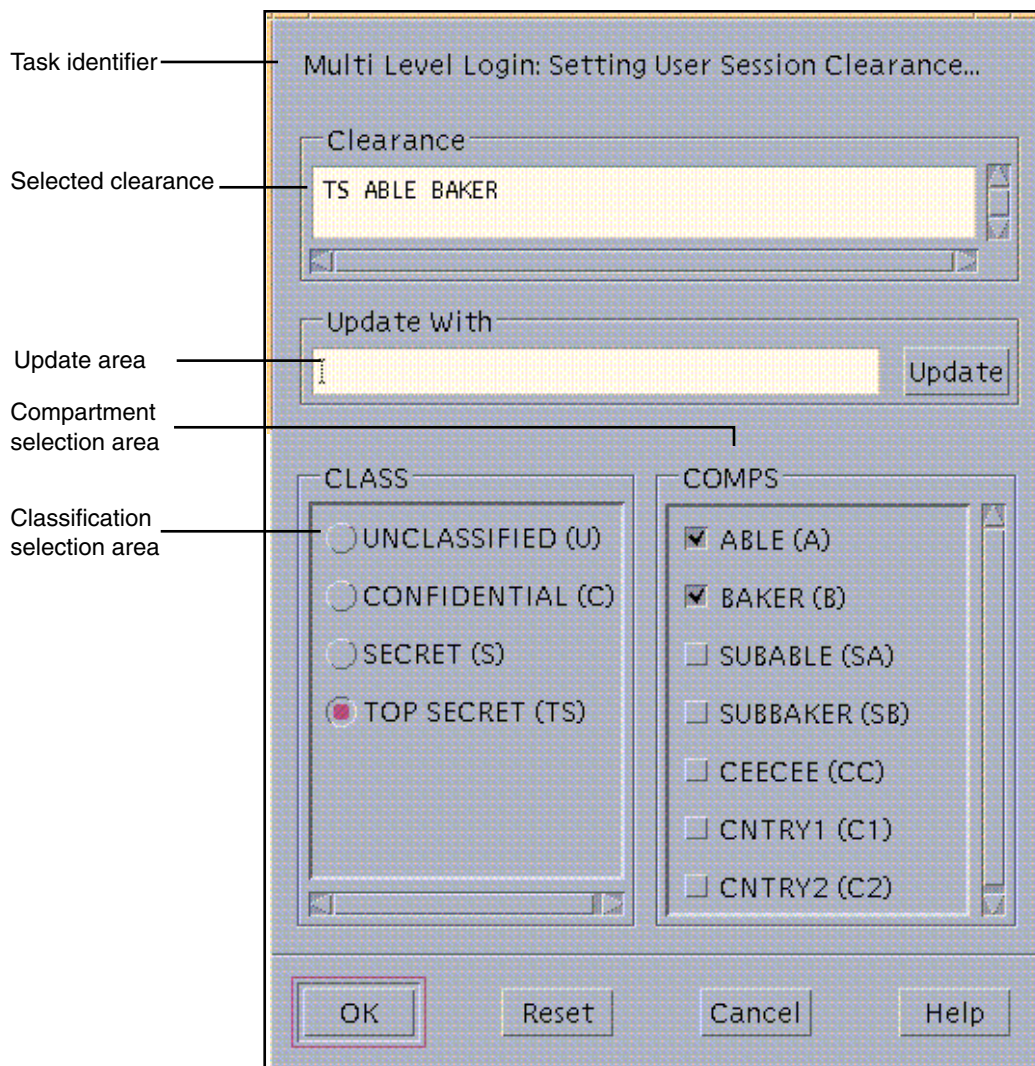
5. Decide whether the session should be single- or multiple-label.

If your account is configured to work at only one label, Single Level Session Label: *Label* appears at the bottom of the dialog box.

- If your account is configured to work at multiple labels and you want to work with multiple labels, proceed to the next step.
- If your account is configured to work at multiple labels but you want to work at only one label, select Restrict Session to a Single Label at the bottom of the dialog box.

6. Press Return or click OK.

- If you can work at only one label or if you restricted the sessions to a single label, the Single-Label Session Login: Setting Session Label dialog box appears.
- If you are allowed to work at multiple labels and decided to do so, the Multilabel Login: Setting Session Clearance dialog box appears.



7. Set the clearance (for a multiple-label session) or the label.

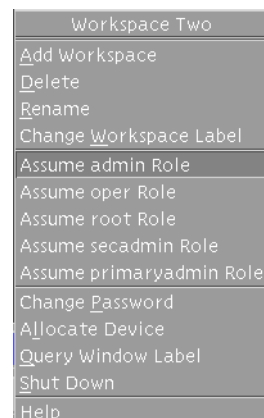
- To accept the default clearance or default label, click OK.
- To specify a different clearance, type the clearance name in the Clearance field or the name of the label in the Update With field.
- To build a clearance (for multiple-label sessions) or label (for single-label sessions) interactively, select a classification in the Class area and the desired compartment components in the Comps area, then click OK.

Note – Other words may be configured at your site to appear instead of Class and Comps. See “Changing Label Component Names on Label Builders” in *Trusted Solaris Label Administration* for information on how the words can be changed.

8. Right-click the center of the Front Panel and choose the Assume *role_account_name* Role option from the Trusted Path (TP) menu.

This option does not appear if you have not been assigned a role.

The following figure shows the Trusted Path menu for a user who is configured to assume the System Administrator role.



9. Type the role password when prompted and click OK.

An administrative role workspace becomes active, and a new administrative role workspace button is added to the workspace switch area.

▼ To Leave an Administrative Role

You can leave an administrative role by choosing a user workspace, deleting the role workspace, or logging off the computer.

- **To choose a user workspace, click the workspace button in the Front Panel.**

The role workspace is still available to you.

- **To delete the workspace, remove all applications from the workspace, then right-click the role workspace button in the Front Panel and choose Delete.**

If others have access to this computer during your session, this procedure prevents them from using your role workspace.

- To log off the computer, click the EXIT button in the Front Panel.

▼ To Launch the Solaris Management Console

The first time on a system that you launch the Solaris Management Console and click the Load button, a delay occurs while the tools are registered and the `/var/sadm/smc/` directory and its subdirectories are created. This delay typically occurs during system configuration.

When a name service is being used, the toolbox with the appropriate scope (either NIS or NIS+) must be edited on the name service master. See “Edit SMC Toolbox Definition for the Name Service” in *Trusted Solaris Installation and Configuration* for how to edit the toolbox.

You also edit the name service toolbox on the clients when you want to be able to modify the NIS maps or NIS+ tables from the client. The procedure “(Optional) Copy the SMC Name Server Toolbox Definition to the Client” in *Trusted Solaris Installation and Configuration* describes how to copy the name service master’s toolbox definition to each client.

Note – Name services support centralized administration of all user, host, and network information, which is important for both user accountability and trusted administration. Administering users and hosts locally is not as secure. There may be special circumstances where a knowledgeable security administrator decides that local accounts are both needed and permissible within your organization’s security policy—even though they can make the system harder to protect and to maintain.

1. Assume a role that is configured to use the Solaris Management Console (SMC) and launch the tool in an administrative role workspace at `ADMIN_LOW` in one of the following ways:

- From the Tools subpanel on the Front Panel, choose the Solaris Management Console option.
- Click the Applications icon on the Applications subpanel of the Front Panel, then double-click the Solaris Management Console icon.
- Invoke the `smc` command in a terminal.
- From the Workspace Menu->Tools submenu, choose the Solaris Management Console option.

2. Select a name from the Server list, or type the name of the computer in the Server field, and then click the Load button.

The term *server* in this context is used to refer to a computer where the SMC server software is running.

The names of any SMC toolboxes on the specified server are loaded into the Toolboxes field.

3. Select the Trusted Solaris Management Console.

4. From the list, choose a Trusted Solaris toolbox of the appropriate scope.

The name of each toolbox starts with the name of the host where the SMC server software is running followed by one of three different scopes (Files, NIS, or NIS+), and then by a policy assignment. For example, the following shows the toolbox with the Files scope and the TSOL Policy for the Server eagle:

eagle: Scope=Files, Policy=TSOL

Note – When you are working in a Trusted Solaris environment, make sure that the Policy=TSOL on the toolbox you select. Only if you were administering a Trusted Solaris system remotely from a Solaris host would you select a toolbox whose Policy=SUSER. If no policy is specified in the toolbox name, the default is SUSER.

Scope Name	Updates
Files	Local files on the current computer.
NIS	NIS maps on the NIS name server for a NIS client host.
NIS+	NIS+ tables on the NIS+ name server for a NIS+ client host.

Note – If you are on a name service client, the name service scope works *only* if you have edited the toolbox files correctly on the client.

5. (Optional) Save the current toolbox to save reloading time:

- a. Choose Console->Preferences.
- b. On the Console tab, click the Use Current Toolbox button.
- c. Click OK.

6. Click the desired SMC tool.

In a name service scope, click Trusted Solaris Configuration to see the “Users” and “Computers and Networks” tools.

In a files scope, click Trusted Solaris Configuration to see the “Users”, “Computers and Networks”, and “Interface Manager” tools.

7. Type the role's password when prompted.

See other chapters in this guide for how to use the Users, Interface Manager, and Computers and Networks tools. Refer to the online help for additional information about the above-named tools and all other SMC tools.

8. When done, choose Exit from the Console menu.

▼ To Launch Local Administrative Actions

- 1. Log in as a user who is able to assume an administrative role and assume the role.**
See "To Log In and Assume a Role" on page 25 if needed.
- 2. Click the Application Manager icon from the Applications subpanel on the Front Panel.**



The Application Manager folder displays.



- 3. Double-click the System_Admin icon in the Application Manager folder.**
- 4. Double-click the icon for the desired administrative action.**

▼ To Edit a Local File

1. **Double-click the Admin Editor action in the System_Admin folder.**
See “To Launch Local Administrative Actions” on page 32 if you have not used the System_Admin folder before.
2. **Type the pathname to the file in the dialog box that appears, and click OK.**
3. **Edit the file using the `adminvi` text-editing commands.**
The `adminvi(1)` man page notes differences between its commands and `vi(1)` commands.
4. **When finished editing, save the changes and quit the file.**

`:wq`

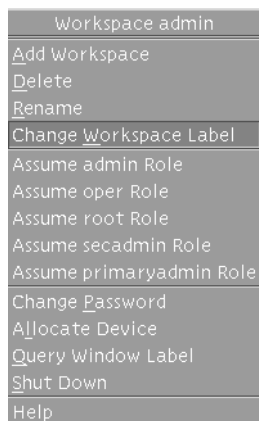
Use `:wq!` if you have difficulty saving a file.

Note – You are not able to save to another file name from within the editor.

▼ To Work at a Different Label

In a multilevel session, working at a different label requires creating a new role workspace and relabeling it.

1. Add a new role workspace by pressing the right mouse button over a role workspace button to bring up the Workspace *Role_name* menu.



2. Choose Add Workspace from the menu.

A new role workspace becomes active, and a new role workspace button appears in the workspace switch area in the Front Panel.



New workspace button

By default, the name of new workspace is the name of the role account followed by an underline followed by a number. As shown in the example, the name of a second administrative workspace created for the admin role is admin_1.

3. Change the label of the workspace by pressing the right mouse button over the new role workspace button and choosing Change Workspace Label.

The Label Builder displays.

4. In the Label Builder dialog box, type the desired label in the text entry field under Update With, click the Update button, and click OK.

The label of the workspace changes to the label you specified in the Label Builder. Windows and applications that were invoked before the label change continue to run at the previous label.

▼ To Enable Any Role to Log In Remotely

See “Managing Remote Logins” on page 67 for a description of the conditions that permit and disallow remote logins.

Note – Do the following on every computer where the role will work, to enable remote logins from that computer.

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
2. Use the Admin Editor action from the System_Admin folder to open the /etc/default/login file for editing.
3. Insert a pound sign (#) to comment out the line: `CONSOLE=/dev/console`.

```
#CONSOLE=/dev/console
```

4. Save and quit the file.

▼ To Log In Remotely From the Command Line

Prerequisite—The role must have the Remote Login authorization, which by default is in two Rights profiles: Remote Administration, and Maintenance and Repair.

1. Ask the Security Administrator to do the procedure “To Enable Any Role to Log In Remotely” on page 35 on every computer you want to use for remote logins.
2. Log in to a computer that the Security Administrator has set up for remote logins, and assume a role.
See “To Enable Any Role to Log In Remotely” on page 35 for the setup procedure.
3. Log in to a remote host by typing `rlogin`, `telnet`, or `ftp` in a terminal in the role’s workspace.

If the `rlogin(1)` or `telnet(1)` command is used to log in, all commands assigned in the current role’s rights profiles are available.

If the `ftp` command is used, see the `ftp(1)` man page for the commands that are available.

▼ To Launch Administrative Actions Remotely

1. Make sure the following prerequisites are satisfied:

- Administrative roles can remotely log in to the computer, as described in “Allowing Remote Logins by Administrative Roles” on page 96
- The `CONSOLE=/dev/console` line in the `/etc/default/login` file is commented out on the current host. The procedure is described in “To Enable Any Role to Log In Remotely” on page 35.
- If administering NIS+ from a NIS+ client, make sure that every NIS+ client’s name is entered in the NIS+ admin group on the domain’s NIS+ master, as described in “To Enable a Role to Administer NIS+” on page 102.

2. Assume an administrative role that either has the `dtappsession` command in one of its rights profiles or that has the authorizations to use the SMC.

Note – The `dtappsession` command is in the Remote Administration profile that is included in the default profiles for all the recommended roles. The command can be launched from an administrative role workspace or can be launched as a Legacy Application in the SMC. In the list of Legacy Applications, you can differentiate the tool for the `dtappsession` command by looking for the Application Manager icon that appears to the left of the words Legacy Application. See the `dtappsession(1)` man page for more information.

3. To use the `dtappsession` command from the SMC, double-click the File Manager icon in the list of tools, and go to step 5.



Legacy Applications

4. To use the `dtappsession` command in a terminal, do the following:

- a. To avoid confusion between the remote CDE applications and any local ones, dedicate an administrative role workspace to this procedure.
See “To Work at a Different Label” on page 33 for how to add an administrative role workspace, if needed.
- b. In the new dedicated workspace, use the `rlogin(1)` command followed by the name of the remote host where you plan to administer.

```
# rlogin e10000domain1
```

- c. **Start remote administration by typing `dtappsession` followed by the name of the local host.**

You can also set `DISPLAY` environment variable on the remote host with the name of the local host. The following screen shows the command entered with the local host name of `ssp_host`.

```
# /usr/dt/bin/dtappsession ssp_host
```

An Application Manager that is running on the remote host displays on the local host.

As shown in the following figure, the `dtappsession` command brings up a Remote Administration dialog box with the name of the remote host followed by the words: Remote Administration. An Exit button displays at the bottom of the screen. The example shows the wording when the remote host's name is `e10000domain1`:

e10000domain1: Remote Administration Press Exit to log out of e10000domain1
Exit

5. **When finished using the remote Application Manager, click the Exit button on the Remote Administration dialog box.**



Caution – Be aware that closing the Application Manager does not end the session.

6. **If you launched the `dtappsession` command from a terminal, exit the remote login session and verify that the terminal is returned to the local host.**

```
$ hostname  
e10000domain1  
$ exit  
$ hostname  
ssp_host
```


Administering Security Requirements

This chapter provides information about notifying users about security, changing security defaults, extending existing security mechanisms, and getting security information. Most of these tasks involve modifying files on a local system. This chapter contains the following procedures:

- “To Change the Allowed Number of Password Tries” on page 53
- “To Prevent Account Locking for Individuals” on page 53
- “To Prevent Account Locking for All User Accounts” on page 54
- “SPARC: To Enable Keyboard Shutdown” on page 54
- “To Prevent Logins From Being Disabled After a Reboot” on page 55
- “To Modify the Selection Configuration File” on page 57
- “To Change Configurable Kernel Switch Settings” on page 56
- “To Add an Authorization to the Environment” on page 57
- “To Add a Privilege to the Environment” on page 59
- “To Get a Hexadecimal Equivalent for a Label” on page 61
- “To List a User’s Home Directory SLDs and Their Labels” on page 61

Enforcing Security Requirements

To ensure that the security of the system is not compromised, administrators need to protect passwords, files and audit data and to train computer users to do their part. To be consistent with the requirements for an evaluated configuration, follow the guidelines in this section.

Training Users About Security Requirements

Each site's security administrator ensures users are trained. The security administrator should hand off the following rules to new employees and remind existing employees of these rules from time to time.

Your organization may wish to provide additional suggestions beyond those listed below.

Users' Security Rules

Anyone who knows your password can access the same information that you can without being identified and therefore without being accountable.

Do not tell anyone else the password.

Do not write the password down or include it in an email message.

Choose passwords that are hard to guess.

Do not leave your computer unattended without locking the screen or logging off.

Be aware that sender information in email can be forged.

Remember that administrators do not rely on email to send instructions to users. Do not ever follow instructions from administrators in an email without first double-checking with the administrator.

Do not send your password to anyone by email.

Because you are responsible for the access permissions on files and directories that you create, make sure that the permissions on your files and directories do not allow unauthorized users to read or change a file or list the contents of or write into a directory.

Using Email

It is poor practice to use email to instruct users to take an action.

Tell users not to trust email with instructions that purport to come from an administrator. This prevents the possibility that spoofed email messages could be used to fool users into changing a password to a certain value or divulging the password, which could subsequently be used to log in and compromise the system.

Enforcing Password Requirements

The System Administrator role is responsible for specifying the original password for each account and for handing off the passwords to new accounts. The System Administrator role must specify a unique user name and a unique user ID when creating a new account. When choosing the name and ID for a new account, the administrator must ensure that both the user name and associated UID are not duplicated anywhere on the network and have not been previously used.

Security Administrator Password Administration Rules

Make sure that the accounts for users who are able to assume the Security Administrator role are configured so that the account cannot be locked. This ensures that at least one account can always log in and assume the Security Administrator role to reopen everyone's account if it ever happens that all other accounts are locked.

Hand over the password to an account in such a way that the password cannot be eavesdropped by anyone else.

Change an account's password if there is any suspicion that the password has been discovered by anyone who should not know it.

Never reuse user names or UIDs over the lifetime of the system.

Ensuring that user names and UIDs are not reused prevents possible confusion over:

- Which actions were performed by which user when audit records are analyzed
- Which user owns which files when archived files are restored

Changing Root's Password

The Security Administrator role can change any account's password at any time except for the password of the root role. Because root's UID 0 is below 100, the SMC considers root to be a "system account," and the SMC does not allow any changes to be made to system accounts. If root's password needs to be changed, root must make the change using the TP menu Change Password option.

Protecting Information

Administrators are responsible for correctly setting up and maintaining DAC and MAC protections for security-critical files, such as the `shadow(4)` file containing encrypted passwords, the local `prof_attr(4)`, `exec_attr(4)`, and `user_attr(4)` databases, and the audit trail.



Caution – Because the protection mechanisms for NIS maps and NIS+ tables are not subject to the access control policy enforced by the Trusted Solaris software, the default NIS maps and NIS+ tables should not be extended, and their access rules should not be modified.

Protecting Passwords

In local files, passwords are protected from viewing by DAC and from modifications by both DAC and MAC. Passwords for local accounts are maintained in the `shadow(4)` file that is readable only by root: The Security Administrator role should ensure that the `/etc/shadow` file is protected by MAC at `ADMIN_LOW`, and by DAC by root (owner), sys (group), and 400.

```
trusted4% ls -l /etc/shadow; getlabel /etc/shadow
-r----- 1 root sys 307 Sep 7 2001 /etc/shadow
/etc/shadow: [ADMIN_LOW]
```

The password field in the NIS+ `passwd.org_dir` table is protected by NIS+ restrictions on access to fields within tables. When any user or administrator tries to view the `passwd.org_dir` table, the only encrypted password that displays is the one belonging to the account.

The following example shows that while user ashish's password field shows as `*NP*` when the user roseanne invoked the `niscat(1)` command, barbar can see the encrypted password for her own account.

```
trusted5% whoami
roseanne
trusted6% niscat passwd.org_dir
. . .
ashish:*NP*:33333:10:Ash Ish:/home/ashish:/bin/csh:*NP*
barbar:0dk1EW44:10:Bar Bara:/home/barbara:/bin/csh:38442:::..:
```

There is no `shadow.org_dir` table.

With NIS, configure the shadow database as a secure map. Secure maps are only readable from a privileged port, thus only a privileged program could access the encrypted password. Sites that need more security than NIS provides should use NIS+.

Administering Groups

The admin needs to verify on the local system and on the network that all groups have a unique group ID (GID).

When a local group is deleted from the system, the administrator role must ensure the following:

- All objects with the GID of the deleted group must be deleted or assigned to another group.
- All users who have the deleted group as their primary group must be reassigned to another primary group.

Deleting Users

When an account is deleted from the system, the administrator and Security Administrator must take the following actions:

- The account's home directory must be deleted.
- Any processes or jobs belonging to the deleted account must be removed:
 - Any objects owned by the account must be deleted or the ownership must be assigned to another user.
 - Any `at` or `batch` jobs scheduled on behalf of the user must be deleted. See the `at(1)` and `crontab(1)` man pages, if needed.
- The user (account) name and UID must be retired and not reused.

Changing Number of Allowable Password Tries

By default, the Trusted Solaris environment allows a maximum of five failed attempts to enter the correct password during a single access attempt. If a user or role account enters the wrong password one time too many during a single attempt, the account is locked. Having such a limit helps forestall brute force attempts to gain access by guessing multiple different passwords.

A count of incorrect passwords entered during a single attempt is kept in the `flag` field of the user or role's entry in the local `shadow(4)` file or in the NIS+ table.

Note – Because NIS does not make the flag field of the shadow database available, a count of failed retries cannot be maintained on a system that relies on the NIS name service. If enforcing a maximum number of failed login attempts is essential to your site's security policy, use either NIS+ or local files.

If the user or role enters the correct password before the count exceeds the maximum, the flag is re-set to zero (0). If an account enters the wrong password one time too many during a single session, the account is locked, as described in the `passwd(4)` man page.

The number of retries allowed applies only for multiple bad passwords entered in sequence in either of the following two occasions:

- When logging into a host
- When re-authenticating oneself in order to change a password or to assume a role.

If an account is ever locked by inadvertent error, the Security Administrator role can open the account by giving the user a new password using the Password tab in the User Accounts tool or by using the appropriate options with the `smuser(1M)` command line interface.

The Security Administrator role can change the RETRIES limit system-wide and can also change whether the limit applies to all users or individual users. Following are the actions that can be taken to change the default:

- The Security Administrator role can change the maximum number of retries to be any number that is consistent with the site's security policy.

The procedure, "To Change the Allowed Number of Password Tries" on page 53, describes how to set the RETRIES value in the local `/etc/default/login` file.
- The Security Administrator role can specify that any individual user's account cannot be locked or can change the default system wide, so that account locking does not occur for anyone. Role accounts, since they do not log in directly, cannot be locked. The entries for individual users take precedence over system-wide entries, which in turn take precedence over the system default.

The procedure, "To Prevent Account Locking for Individuals" on page 53, describes how to update a user's account with the User Accounts tool to prevent the account from being locked. The procedure, "To Prevent Account Locking for All User Accounts" on page 54, describes how to set a system-wide password lock policy.

Managing the Relabeling of Files

By default, normal users can perform cut and paste, copy and paste, and drag and drop operations on both files and selections *as long as the source and destination have the same label and have the same user ID*.

The `/usr/dt/config/sel_config` file is consulted to determine which actions will be taken when an operation would upgrade or downgrade a label. (The comments and keywords in the file use the terms sensitivity label and label interchangeably.)

Note – The rules that apply when some operations are performed on file icons differ from the rules that apply when the same operations are performed on selections made in windows. Drag and drop of *selections* always requires equality of labels and ownership.

The `sel_config` file defines:

- A list of selection types to which automatic replies are given
- Whether certain types of operation should be automatically confirmed or
- Whether a selection confirmer dialog should be displayed

The following figure shows the selection confirmer for drag and drop operations between File Managers. Other slightly-different selection confirmers display for cut and paste and copy and paste operations between File Managers and between windows at varying labels.

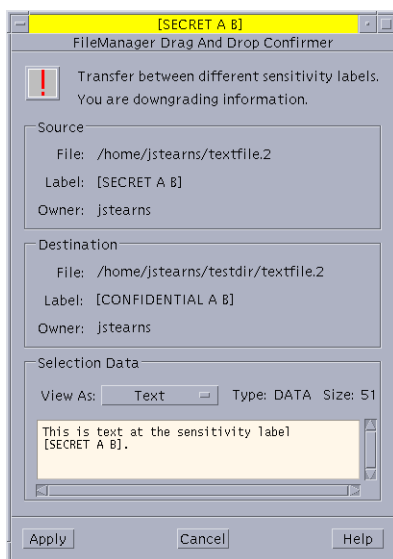


FIGURE 2-1 File Manager Selection Confirmer

The Security Administrator role can change the defaults by using the Selection Configuration action. The new settings become effective the next time anyone logs in.

Users can copy and paste between file managers that they own and that are at the same label. The types of operations that may be performed on files with varying label and ownership relationships are summarized and shown with the authorizations needed, in the following table.

TABLE 2-1 Conditions for Moving Files Between File Managers

Transaction Description	Label Relationship	Owner Relationship	Authorization(s) Required
Copy/Cut and paste, or drag and drop of files between File Managers	Same label	Same UID	None required
	Downgrade	Same UID	Downgrade file label
	Upgrade	Same UID	Upgrade file label
	Downgrade	Different UIDs	Downgrade file label Act as file owner
	Upgrade	Different UIDs	Upgrade file label Act as file owner

Users can copy and paste between *windows* that they own and that are at the same label. The types of operations that may be performed on selections between windows

with varying label and ownership relationships are summarized and shown with the authorizations needed in the following table.

TABLE 2-2 Conditions for Moving Selections Between Windows

Transaction Description	Label Relationship	Owner Relationship	Authorization(s) Required
Copy/Cut and paste of selections between windows	Same label	Same UID	None required
	Downgrade	Same UID	Paste to a downgraded window
	Upgrade	Same UID	Paste to an upgraded window
	Downgrade	Different UIDs	Paste to a downgraded window Act as file owner
	Upgrade	Different UIDs	Paste to an upgraded window Act as file owner
Drag and drop of selections between windows	Same SL always required	Same UID always required	None applicable

sel_config File Sections

The rules in the `sel_config` file apply to cut and paste, copy and paste, and drag and drop of files between file managers. (See `dtfile(1)` and the *Trusted Solaris User's Guide* for more about the File Manager application.) The rules in the `sel_config` file also apply to cut and paste and copy and paste between windows. Drag and drop between windows is mediated by the `/usr/dt/bin/sel_mgr` application, not by `sel_config`.

The `sel_config` file has two sections described below:

- Automatic confirmation
- Automatic reply

Automatic Confirmation Section

The format of each line in the automatic confirmation section of the `sel_config` file is shown in the following table. *label-relation* refers to the relationship between the label of the source and the label of the destination, and the value `n` means to display the selection confirmer to the user.

Transfer Type	Automatically confirm?
<i>label-relation</i> (upgrade downgrade equal disjoint)	y n

Automatic Reply Section

The `autoreply` field defines the type of reply for all the named types of selections that follow it. This section provides a way to reply automatically to several types of selections at once instead of having to respond to each individually. See the `sel_config(4)` man page for more information.

Extending Authorizations and Privileges

The following Trusted Solaris security mechanisms are extendable:

- **Audit events and classes**—Adding audit events and audit classes is described in the *Trusted Solaris Audit Administration*.
- **Rights profiles**—Adding rights profiles is described in “Adding or Modifying a Rights Profile” on page 87.
- **Roles**—Adding roles is described in “Creating a New Role” on page 96.
- **Authorizations and privileges**—The rest of this section describes how to add authorizations and privileges.

Adding New Authorizations

Adding a new authorization consists of:

1. Adding a header entry for the site’s authorizations into the `auth_attr(4)` database.
2. Adding a grant authorization into the `auth_attr` database that enables a role to assign the new authorization to others.
3. Adding the new authorization entry to the `auth_attr` database.
4. If you are running a name service, adding the new entries to the name service `auth_attr` database.
5. Writing or modifying an application to check for the new authorization.

In a default Trusted Solaris system, only the device allocation mechanism accepts new authorizations. Of course, a site can write other applications that check for new authorizations.

The example detailed in “To Add an Authorization to the Environment” on page 57 makes use of the fact that the device allocation authorization is configurable.

6. Assigning the new authorization to user or role accounts.

The format for an entry in the `auth_attr(4)` file is:

```
name:res1:res2:short_desc:long_desc:attr
```

The `short_desc` field is a brief description of the activity permitted by the authorization. The `long_desc` is used by the Solaris Management Console when it displays authorizations. A help file, which is specified in the `attr` field using the keyword value pair `help=filename`, displays in the online help. *filename* must be located in the directory ending with the name of the locale:
`/usr/lib/help/auths/locale/localename`.

The following screen shows the default device allocation authorization in the `auth_attr` file in the C locale. The help file in the C locale is
`/usr/lib/help/auths/locale/C/DevAllocate.html`.

```
solaris.device.allocate:::Allocate Device::help=DevAllocate.html
```

The example below shows two finer-grained device allocation authorizations that could be used to replace the default one above, one for tape devices and one for floppy devices. In the example, the authorizations' names start with the Internet domain name of the NewCo company.

```
com.newco.device.allocate.tape:::Allocate Tape Device::help=TapeAllocate.html
com.newco.device.allocate.floppy:::Allocate Floppy Device::help=FloppyAllocate.html
```

The next example shows the `solaris.allocate.device` authorization replaced in the `device_allocate(4)` file entry for `floppy_0` with `com.newco.device.allocate.floppy`. This change would be made by the Security Administrator role using the Device Allocation Manager, as described in "To Add an Authorization to the Environment" on page 57. After this substitution, any user attempting to allocate the floppy device must have the new authorization.

```
floppy_0;fd;0x0000000000000000000000000000000000000000000000000000000000000000
000;0x7fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff;com.
newco.device.allocate.floppy;/etc/security/lib/disk_clean
```

Adding New Privileges

Adding a new privilege consists of adding an entry for the privilege into these two files:

- `/usr/include/sys/tsol/priv_names.h`
- `/usr/lib/tsol/locale/C/priv_name`

The `priv_names.h` File

The `/usr/include/sys/tsol/priv_names.h` header file contains manifest constants and associated numbers for privileges. Up to 128 possible privileges are allowed. As shown in the following screen example, the definitions for the default privileges range from 1 to 86 (with 0 meaning no privileges). Not all 86 privileges are defined since some have been retired.

The manifest constants and numbers for default privileges in `priv_names.h` are:

```
PRIV_FILE_AUDIT = 1,          /* operational */
PRIV_FILE_CHOWN = 2,         /* operational */
PRIV_FILE_DAC_EXECUTE = 3,   /* policy */
.
.
.
PRIV_WIN_SELECTION = 84,     /* operational */
PRIV_WIN_UPGRADE_SL = 86,    /* operational */
```

Privileges available for extension follow the `/* Reserved for ISV... */` text in the file:

```
/* Reserved for ISV, GOTS, integrator, ... use */
.
.
reserved127 = 127,
reserved128 = 128
```

Note – If you wish to interoperate with other systems, you should contact your Trusted Solaris representative to reserve a privilege number.

The `priv_name` File

The following is the format for an entry in `/usr/lib/tsol/locale/locale_name/priv_name`:

number : *name* : *description*

The value of *number* in the `priv_name(4)` file must match the privilege ID in the `/usr/include/sys/tsol/priv_names.h` file. *name* must be concise and descriptive for display in user interfaces.

description describes the activity permitted by the privilege. The definition guides the Security Administrator role when assigning privileges to programs.

The following is an example of a privilege in the default `priv_name` file:

```
4:file_dac_read:Allows a process to read a file or directory \
whose permission bits or ACL do not allow the process read permission.
```

Changing CDE Defaults

In the default CDE environment, users can add actions to the Front Panel and customize the Workspace menu. Trusted Solaris software limits users' ability to add programs and commands to the CDE.

Customizing the Workspace Menu

The Workspace Menu is the menu accessed by clicking and holding the right mouse (Menu) button on the background of the workspace. Using the Customize Menu and Add Item to Menu options on the Workspace Menu is the same as in the base CDE window system, with some Trusted Solaris protections.

The following apply when a user is allowed to work at multiple labels:

- The user must use the Customize Menu and Add Item to Menu options in a workspace labeled at the session clearance. Changes made at other labels than the session clearance are not recognized by the window system.

If a user is able to log in at multiple labels, the user has the potential for multiple session clearances during different login sessions. Therefore, make any changes at each of the potential session clearances if you want the changes to apply to all potential login sessions.

- The user makes the changes in a normal user workspace.
- When the user assumes a role, changes to the Workspace Menu persist.
- Changes made to the Workspace Menu are stored in the user's home directory in the single-level directory (*SLD*) created at the working label. The label should be the same as the session clearance. The items in the Workspace Menu are stored in the `.dt/wsmenu` directory within the user's multilevel (*MLD*) home directory in the *SLD* that corresponds to the working label.

For example, to change the Workspace Menu when the user's only possible session clearance is `NEED_TO_KNOW ENG`, the user would go to a workspace labeled `NEED_TO_KNOW ENG`. If the user adds an item to the Applications menu using the Add Item to Menu option, the item would be stored in `/home/username/.dt/wsmenu/Applications`.

The pathname above corresponds to the real *MLD* path shown below, where `.SLD.3` in the example is the *SLD* that corresponds to the `NEED_TO_KNOW ENG` label for user `barbar`.

`/home/.MLD.barbar/.SLD.3/.dt/wsmenu/Applications`

- The profile mechanism must enable the user to run the action.

Any option added to the Workspace Menu must be handled by one of the user's rights profiles or the option will fail when invoked and an error message will display.

For example, anyone with the Run action can double-click the icon for any executable and run it, even if the action or any commands it invokes are not in one of the account's rights profiles. By default, roles do not have the Run action, and all executable actions require the Run action, and therefore, any item that requires the Run action fails when executed by a role.

Customizing the Front Panel

Anyone can drag and drop a pre-existing action from the Application Manager to the Front Panel as long as the account doing the modification has the action in its profile. Actions in the `/usr/dt/*` or `/etc/dt/*` directories can be added to the Front Panel, but applications in the `$HOME/.dt/appconfig` directories cannot. While users can use the Create Action action, they cannot write into any of the directories where the system-wide actions are stored, so they cannot use the actions.

In the Trusted Solaris environment, the actions' search path has been changed so that actions in any individual's home directory are processed last instead of first. Therefore, no one can customize existing actions.

The Security Administrator role has the Admin Editor action, so can make any needed modifications to the `/usr/dt/appconfig/types/C/dtwm.fp` file and the other configuration files for the Front Panel subpanels. This guide contains two procedures that exemplify how to modify existing files to create new actions. "To Add Actions Outside of the System_Admin Folder" on page 248 describes how to create an alternate mail application that can run with privilege in the Front Panel. "dtmail is the Default Mail Application" on page 107 describes how to add an administrative action that can run with inherited privileges to the System_Admin folder for the purpose of editing another configuration file.

Roles can drag and drop actions from the System_Admin folder to the Front Panel. The icons can confuse normal users because the action icons only work for the roles.

Changing and Accessing Security Information (Tasks)

▼ To Change the Allowed Number of Password Tries

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
2. Use the Admin Editor action to open the `/etc/default/login` file for editing.
See “To Edit a Local File” on page 33, if needed.

3. Search for the string `#RETRIES`.

```
# RETRIES sets the number of consecutive authentication failures
# allowed before the user is locked out.
#
#RETRIES=5
```

4. Change the `RETRIES` value to the desired value and remove the `#` sign.
5. Save and close the file.

▼ To Prevent Account Locking for Individuals

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
2. Bring up the Solaris Management Console on the desired server with the desired scope, either Files, NIS, or NIS+.
See “To Launch the Solaris Management Console” on page 30, if needed, for how to bring up the Solaris Management Console.
3. Bring up the User Accounts tool by clicking the Trusted Solaris Configuration icon, then clicking the Users icon. Enter the role password when prompted.
4. If the user account does not exist, create it by double-clicking the User Accounts icon in the Navigation pane, clicking Add User in the Action menu, and then choosing With Wizard or From Template.

Follow the instructions in the help text for how to fill in the fields to add a user account.

5. Open the User Properties tool by double-clicking the User Accounts icon, then double-clicking the name of the user.
The Properties dialog displays.
6. Click the Trusted Solaris Attributes tab.
7. In the Account Usage section, select **No** from the pull-down menu next to **Lock** account after maximum failed logins.

▼ To Prevent Account Locking for All User Accounts

1. Assume the Security Administrator role and go to an **ADMIN_LOW** workspace.
See “To Log In and Assume a Role” on page 25, if needed.
2. At **ADMIN_LOW**, use the Admin Editor action to open the `/etc/security/policy.conf` file for editing.
See “To Edit a Local File” on page 33, if needed.
3. Search for the string `LOCK_AFTER_RETRIES`.
`LOCK_AFTER_RETRIES=yes`
4. Change **yes** to **no**, or if the string is not present in the file, add the following.
`LOCK_AFTER_RETRIES=no`
5. Save and quit the file.
`:wq`

▼ SPARC: To Enable Keyboard Shutdown

By default, Trusted Solaris systems can only be brought down by an orderly shutdown through the Shut Down option from the Trusted Path menu. The Stop–A keyboard shutdown does not work.

Where the site’s security policy allows, the default setting can be changed. On hosts that are used by administrators for debugging, this setting can be changed to allow access to the `kadb(1M)` kernel debugger.

1. Assume the Security Administrator role and go to an **ADMIN_LOW** workspace.
2. Use the Admin Editor action to open the `/etc/default/kbd` file for editing.
See “To Edit a Local File” on page 33, if needed.

3. Search for the string `KEYBOARD_ABORT`.

```
KEYBOARD_ABORT=disable
```

4. Enter a pound sign at the start of the line to comment it out.

```
#KEYBOARD_ABORT=disable
```

▼ To Prevent Logins From Being Disabled After a Reboot

In the Trusted Solaris environment, the `/etc/nologin` file is created after boot and is not removed until a user with the Enable Logins authorization enables logins.

If your site's security policy allows, the Security Administrator role can edit the `RMTPFILES` script in `/etc/init.d` to comment out the lines that recreate the `/etc/nologin` file. See "To Prevent Logins From Being Disabled After a Reboot" on page 55, if changing the default is consistent with your site's security policy.

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.
2. Use the Admin Editor action to open the `/etc/init.d/RMTPFILES` for editing.
See "To Edit a Local File" on page 33, if needed.

Note – Do not create a backup file in the `/etc/init.d` directory. Because all files in the startup directories are executed, the backup file would be executed after the changed version, so the `/etc/nologin` file would be re-created, and the effect of this procedure would be undone.

3. Comment out the lines that disable logins after a reboot.

Comment out the active lines as shown in the following screen.

```
# cp /dev/null /etc/nologin
# echo "" >> /etc/nologin
# echo "NO LOGINS: System booted" >> /etc/nologin
# echo "Logins must be enable by an authorized user." >>
# /etc/nologin
# echo "" >> /etc/nologin
```

4. Save and quit the file.

```
:wq
```

▼ To Change Configurable Kernel Switch Settings

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
2. Use the Admin Editor action from the System_Admin folder in the Application Manager to open `/etc/system` for editing.
3. Set the configurable switches as desired, then save the file.
See Table 2–3 for a description of the switches.
4. Reboot the system for the values to go into effect.

The following table shows the customer-configurable kernel switches in the `system(4)` file.

TABLE 2-3 Configurable Trusted Solaris Switches in `/etc/system`

<code>tsol_admin_high_to_cipso</code>	The <code>tsol_admin_high_to_cipso</code> switch is not in the default <code>/etc/system</code> file, but it can be added if needed. The default setting in the kernel is 0. To enable communications with TSIX-type hosts that have the IP Label Field specified as CIPSO, this switch must be set to 1. This causes the label on a packet to be mapped to a valid CIPSO label with the highest classification and all compartments turned on, instead of being dropped. See “CIPSO Labels in Packets” on page 132 for more information.
<code>tsol_clean_windows</code>	To support object reuse, the <code>tsol_clean_windows</code> switch is set to 1 by default, to clear inactive register windows on return from each system call. Setting the switch to 0 disables the cleaning of inactive windows after each system call, allowing the possibility that a system call can return kernel information from an inactive register window.
<code>tsol_flush_buffers</code>	Between the time when blocks are linked to an inode and written to disk, a crash could leave old disk blocks (possibly of a higher label) linked to a file system after <code>fsck(1M)</code> recovers the file system. To ensure that data blocks are flushed before inodes are updated on disk, the <code>tsol_flush_buffers</code> switch is set to 1 by default. There is a small performance penalty. Setting this switch to 0 disables the forced data flushing before inode updates.

TABLE 2-3 Configurable Trusted Solaris Switches in `/etc/system` (Continued)

<code>tsol_hide_upgraded_names</code>	<p>Actions by users with the Upgrade File Label authorization and by processes with the <code>file_mac_write</code> and <code>file_upgrade_sl</code> privileges can either create a new file or subdirectory or relabel an existing file or subdirectory at a label that dominates the label of the containing directory. Such files and subdirectories are said to be upgraded and the names of the upgraded files and subdirectories are referred to as <i>upgraded names</i>.</p> <p>At sites that consider upgraded names to be sensitive information, the <code>tsol_hide_upgraded_names</code> switch enables the Security Administrator role to hide upgraded names. Setting this flag prevents <code>getdents(2)</code> from returning upgraded file names. Because all directory entries are examined before the results are returned, there is a performance penalty. Upgraded names display by default.</p>
<code>tsol_privs_debug</code>	<p>The <code>tsol_privs_debug</code> switch allows the administrative use of <code>runpd(1M)</code> to characterize a program's use of privilege. See Chapter 13 under "To Find Out Which Privileges a Program Needs" on page 252 for the complete setup procedure. After the application(s) have been privileged debugged, this variable should be reset and the machine rebooted. Privilege debugging is disabled by default.</p>

▼ To Modify the Selection Configuration File

1. Assume the System Administrator role and go to an **ADMIN_LOW** workspace.
2. Go to the **System_Admin** folder in the Application Manager.
3. Double-click the Selection Configuration action to open the `sel_config` file for editing.
See "Managing the Relabeling of Files" on page 45 for what the fields mean, if needed.
4. Save the changes.
The settings go into effect at the next login.

▼ To Add an Authorization to the Environment

1. Log in and assume the Security Administrator role and go to an **ADMIN_LOW** workspace.

2. Use the Admin Editor action in the System_Admin folder in the Application Manager to open the `auth_attr` file for editing.

Note – If you are using a name service, you need to make the changes in this procedure to the `auth_attr(4)` file in the location from which you populate the entries to the NIS map or NIS+ table. See the *Solaris Naming Administration Guide* for how to populate the name service databases with the new entries.

3. Create a heading for the new authorizations, using the reverse-order Internet domain name of your organization followed by optional additional arbitrary components. Separate components by dots. End heading names with a dot.

The example shows a heading constructed for a company whose Internet domain name is `newco.com`. The name of the company is followed by a dot (`.`).

```
com.newco.::NewCo Header::help=NewCo.html
```

4. Add new authorization entries.

The example shows the authorization to grant all NewCo authorizations, followed by the authorization to grant NewCo's device authorizations, followed by a new tape device allocation authorization, followed by a new floppy device allocation authorization.

```
com.newco.grant::Grant All NewCo Authorizations::  
help=GrantNewco.html  
com.newco.grant.device::Grant NewCo Device Authorizations::  
help=GrantNewcoDevice.html  
com.newco.device.allocate.tape::Allocate Tape Device::  
help=TapeAllocate.html  
com.newco.device.allocate.floppy::Allocate Floppy Device::  
help=FloppyAllocate.html
```

Enter the authorizations one per line. The lines above are split here to fit on the page.

5. Save and close the file.

```
:wq
```

6. If you are using a naming service, update the `auth_attr` NIS map or NIS+ table.

See the `nistbladm(1)` man page and the *Solaris Naming Administration Guide* for how to update the `auth_attr(4)` map or table.

7. Add the authorization to the database that defines which authorizations the application requires.

For example, to assign the new device allocation authorization, you would use the Device Allocation Manager. See "To Add Site-Specific Authorizations to a Device" on page 220 for the procedure.

8. Use the Rights tool to add the new authorizations to the Custom *rolename* Role, and make sure the Custom *rolename* Role rights profile is assigned to the role.

▼ To Add a Privilege to the Environment

Note – Before you add a privilege, contact your Trusted Solaris representative to reserve a privilege number.

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.
2. Use the Admin Editor action to open the `/usr/include/sys/tsol/priv_names.h` file for editing.
See “To Edit a Local File” on page 33, if needed.
3. Follow the directions in the comment at the top of the `priv_names.h` file, shown below.

```
/ *
* ***** IMPORTANT *****
*
* The privilege names should be maintained in alphabetical order
* not numeric order.
*
* When a privilege is retired it should be placed in the appropriate
* reserved area in the form "tsol_reserved## = ##," or
* "reserved## = ##".
*
* When a new privilege is needed, it should be taken from the first
* available privilege in the appropriate reserved area.
*
* ISVs, GOTS', integrators who need privileges are encouraged to
* request and retire them by contacting their respective Trusted
* Solaris support representative.
*
* This file is parsed by the priv_to_str(3) functions.
*
* In order to guarantee correct parsing, the format of the
* following priv_t definition must be preserved.
*
* Specifically, the following guidelines must be followed:
*
* 1. All privileges must have an explicitly assigned id.
*    DO NOT RELY ON COMPILER TO ASSIGN IDs.
*
* 2. One privilege id assignment per line.
*    DO NOT CONCATENATE OR BREAK LINES.
*
* 3. Do not use the '=' character at anywhere other than
*    the privilege id assignment.
*    For example, DO NOT use '=' in the comments.
```

4. Create an entry in the `priv_names.h` file with the manifest constant for the privilege.

A sample entry is below.

```
PRIV_RISKY = 90,
```

5. Save and close the file.

6. Use the Admin Editor action to open the `/usr/lib/tsol/locale/locale_name/priv_name` file for editing.

In the C locale, for example, you would edit the `/usr/lib/tsol/locale/C/priv_name` file.

7. Create an entry with the privilege ID, name, and definition for the privilege in the `priv_name` file.

Note – Make sure that you use the correct privilege ID.

A sample entry is below.

```
90:override everything:Allows a process to bypass all MAC and \
DAC checks and auditing flag settings and be otherwise totally \
unaccountable.
```

8. Save and close the file.
9. Copy the changed `priv_names.h` and `priv_name` files or make the same change in these files on all computers in the Trusted Solaris network.

▼ To Customize the Workspace Menu

Note – To make the changes apply to every possible login session, users with multiple labels need to repeat these steps at every label that corresponds to a potential session clearance.

1. Log in and go to a workspace whose label is the same as the session clearance—without assuming a role.
2. Choose the Customize Menu or Add Item to Menu option from the Workspace Menu and make the desired changes.

▼ To Get a Hexadecimal Equivalent for a Label

Use `atohexlabel(1M)` with the options shown in the following steps to get the hexadecimal equivalents of a label or clearance. The `atohexlabel(1M)` command is in the default Object Label Management profile, which is assigned by default to the Security Administrator role.

1. Assume the Security Administrator role and create an ADMIN_HIGH workspace.

To get the hexadecimal value for a sensitivity label, use `atohexlabel` with the `-s` option followed by the name of the sensitivity label.

[illegible]

2. To get the hexadecimal value for a clearance label, use `atohexlabel` with the `-c` option followed by the name of the clearance.

[illegible]

▼ To List a User's Home Directory SLDs and Their Labels

1. Assume the System Administrator role and go to an ADMIN_HIGH workspace.

2. Find the full pathname of the user's home directory MLD.

```
% mldrealpath /home/janez
/home/.MLD.janez/.SLD.3:
% getlabel /home/.MLD.janez/.SLD.*
/home/.MLD.janez/.SLD.0:      [ADMIN_LOW]
/home/.MLD.janez/.SLD.1:      [CONFIDENTIAL]
/home/.MLD.janez/.SLD.2:      [TOP SECRET A B]
/home/.MLD.janez/.SLD.3:      [ADMIN_HIGH]
/home/.MLD.janez/.SLD.4:      [SECRET]
/home/.MLD.janez/.SLD.5:      [CONFIDENTIAL A B]
/home/.MLD.janez/.SLD.6:      [SECRET A B]
/home/.MLD.janez/.SLD.7:      [TOP SECRET]
/home/.MLD.janez/.SLD.8:      [UNCLASSIFIED]
```


Managing User Accounts

This chapter describes essential decisions to make before creating regular users, and provides additional background for managing user accounts. The chapter assumes that the install team has set up a limited number of user accounts. These users can assume the roles that configure and administer the Trusted Solaris operating environment. See *Trusted Solaris Installation and Configuration* for details.

This chapter includes the following procedures:

- “To Modify Default User Label Attributes” on page 77
- “To Modify `policy.conf` Defaults” on page 77
- “To Set Up Startup Files for Users” on page 77
- “To Invoke `.login` or `.profile` During Login” on page 79
- “To Force `dtterm` to Launch New Shells as Login Shells ” on page 80
- “To Customize Shell Initialization Files for Users” on page 80
- “To Enable a User to Track Others’ Jobs on a System” on page 81
- “To Enable a User to Track All Others’ Jobs” on page 81

Setup Before Creating User Accounts

The following decisions and setup affect what users can do in a Trusted Solaris environment and how easy it is for them to do it. Some decisions are the same as those you would make when installing a network of the Solaris software or other UNIX systems. However, there are decisions specific to the Trusted Solaris environment that affect site security and ease of use.

Decisions to Implement Before Creating Users

- Decide whether to change default user security attributes in the `policy.conf(4)` and the `label_encodings(4)` files. See “Managing Default User Security Attributes” on page 65 for a description of the defaults.
- Decide which startup files, if any, should be copied or linked from each user’s minimum-label home directory SLD to the user’s higher SLDs. See “To Set Up Startup Files for Users” on page 77 for the procedure.
- Decide whether to permit sourcing of shell initialization files and what content you want to provide. See “Managing Initialization Files” on page 67.
- Decide whether to allow users to remotely log in from the command line. Users who are permitted to remotely log in from the command line require a rights profile that contains the appropriate authorization. The machines themselves must also be enabled for remote login. See “Managing Remote Logins” on page 67 for a discussion of remote logins.
- Decide if users can access peripheral system devices, like the microphone, CD-ROM drive, and tape drive.

If access is permitted to some users, decide if your site requires additional authorizations to satisfy site security. See “Authorizing Device Allocation” on page 210 for the default list of device-related authorizations. “Adding New Authorizations” on page 48 describes a finer-grained set of device authorizations, and “To Add an Authorization to the Environment” on page 57 describes how to implement the new authorizations.
- Decide whether to control a device differently when it is allocated remotely or locally. See Table 12–2 for an example of handling remote and local device allocation differently. The example requires the Security Administrator to create new authorizations.

Decisions to Implement Before Users Log In

- Decide whether devices should be deallocated when the allocating user logs out or reboots the system. See “Configuring a Device” on page 207 for more information.
- Decide whether to change the default security provisions in mail. See “Managing Trusted Solaris Mail Features” on page 105 for mail setup that is most easily done before users log in and use mail.
- Decide whether to hide filenames whose label is higher than the label of the directory that contains the file. See “To Change Configurable Kernel Switch Settings” on page 56 for how to change the switch that controls whether these upgraded file names are visible.

Managing Default User Security Attributes

Settings in the `policy.conf(4)` and the `label_encodings(4)` files together define default Trusted Solaris security attributes for user accounts. The values set explicitly in a user template override these values. Some of the values set in these files also apply to role accounts. The User Template default values are described in detail in “Adding or Modifying a User Account” on page 84.

Label Encodings File Defaults

The `label_encodings` file defines the Minimum Label, Clearance, and Default Label View that are applied to a user account if the attributes are not explicitly set for the account. The values shown in the following table are those in the Trusted Solaris version of the `label_encodings` file. Typically, a site replaces the Trusted Solaris version during system configuration with a site version.

TABLE 3–1 Security Defaults for Users in the `label_encodings` File

Trusted Solaris Attribute	Keyword in LOCAL DEFINITIONS Section	Default
Minimum Label	Default User Sensitivity Label= u;	In ACCREDITATION RANGE Section: minimum sensitivity label=u;
Clearance	Default User Clearance= c;	In ACCREDITATION RANGE Section: minimum clearance= c nationality: cntry1/cntry2;
Default Label View	Default Label View is External;	External

At some sites the names of administrative labels are considered to be classified information. The value `EXTERNAL` hides that classified information.

The user account’s clearance and minimum label must be dominated by the highest label and must dominate the minimum clearance that are defined in the user `ACCREDITATION RANGE` section in the `label_encodings(4)` file. See *Trusted Solaris Label Administration* for more about labels.

The following algorithm determines which value the system uses:

1. If the administrator explicitly set a value in the Solaris Management Console when creating the user, use that value.
2. Otherwise, use the values for the “Default User ...” and “Default Label View” keywords in the `label_encodings` file.
3. If there is no specific value for the “Default User ...” and “Default Label View” keywords, use the Accreditation Range values.

policy.conf File Defaults

The following table shows the default settings in the `policy.conf` file.

TABLE 3–2 Security Defaults for Users and Roles in the `policy.conf`

Attribute	Keyword with Default Setting	System Default
authorizations (from <code>auth_attr(4)</code> database)	<code>#AUTHS_GRANTED=</code>	none
idle action: logout lock	<code>IDLECMD=lock</code> (applies to users only)	lock
idle time: 1 – 120 minutes or Forever	<code>IDLETIME=30</code> (applies to users only)	30 minutes
show or hide labels: <code>hidesl</code> <code>showsl</code>	<code>LABELVIEW=showsl</code>	showsl
lock after bad password limit is exceeded: yes no	<code>LOCK_AFTER_RETRIES=yes</code>	yes
method of password generation: manual auto	<code>PASSWORD>manual</code>	manual
profiles (from <code>prof_attr(4)</code> database)	<code>PROFS_GRANTED=</code>	Basic Solaris User

So, users by default are authorized to view SMC data and to edit their own cron jobs; their system locks after 30 minutes of no activity; they can see the label that they are working in; they will not be able to log in if they fail to provide the correct password for three consecutive tries; they must type in a new password (possibilities will not be generated for them); and they can execute all commands and actions on the system without privilege.

The authorizations (`AUTHS_GRANTED`) and rights profiles (`PROFS_GRANTED`) that are defined in this file are *in addition* to any authorizations and profiles assigned to individual accounts. For the other fields, the following algorithm determines which value the system uses:

1. If the administrator explicitly set a value in the Solaris Management Console when creating the user, use that value.
2. Otherwise, use the value in the `policy.conf` file.

Managing Remote Logins

A remote login between two Trusted Solaris hosts is considered to be an extension of the current login session. An authorization is not required when:

- The user chooses the Remote Login option on the CDE login screen.
- The `rlogin` command does not prompt for a password.

If an `/etc/hosts.equiv` or a `.rhosts` file in the user's home directory on the remote host lists either the username or the host from which the remote login is being attempted, no password is required. See the `rhosts(4)` and `rlogin(1)` man pages for more information.

For all other remote logins, including logins with the `telnet(1)` command, the Remote Login authorization is required.

See "To Create a Rights Profile" on page 89 for how to create and see "To Assign an Authorization to a User" on page 93 for how to assign a new profile to a user.

Managing Initialization Files

Administrators who are setting up shell initialization files must consider certain details that are either not as important in standard UNIX systems or do not apply. The differences exist because of the following aspects of the Trusted Solaris implementation:

- Home directories are *multilevel directories* (MLDs).
- A profile shell can be used to restrict an account's access to commands.
- The execution of the `profile(4)` file is restricted in the Trusted Solaris environment.

As in the Solaris environment, files in the skeleton directory are copied into the user's home directory. However, in the Trusted Solaris environment, a user has a home directory for every label that the user works in. So, files from the skeleton directory are copied to the user's first home directory *single-label directory* (SLD).

A user's first SLD is created by the administrator during account creation. The first SLD is at the label of the creating process, ADMIN_LOW. The following example shows a user's directory structure after first login, and then the user's home directory structure after working at different labels.

EXAMPLE 3-1 User Home Directory Structure after First Login

```
/home/.MLD.janez/.SLD.0:      [ADMIN_LOW]
/home/.MLD.janez/.SLD.1:      [CONFIDENTIAL]
```

EXAMPLE 3-2 User Home Directory Structure after Working at Different Labels

```
/home/.MLD.janez/.SLD.0:      [ADMIN_LOW]
/home/.MLD.janez/.SLD.1:      [CONFIDENTIAL]
/home/.MLD.janez/.SLD.2:      [TOP SECRET A B]
/home/.MLD.janez/.SLD.3:      [SECRET]
/home/.MLD.janez/.SLD.4:      [CONFIDENTIAL A B]
/home/.MLD.janez/.SLD.5:      [SECRET A B]
/home/.MLD.janez/.SLD.6:      [TOP SECRET]
/home/.MLD.janez/.SLD.7:      [UNCLASSIFIED]
```

To copy or link the startup files into other labeled home directories requires setup on the part of the Security Administrator role. See "To Set Up Startup Files for Users" on page 77 for the procedure.

This section provides the background information needed to understand how startup files are administered in the Trusted Solaris environment. Also see the man pages for the `csch(1)`, `ksh(1)`, `sh(1)`, and `pfexec(1)` commands.

A set of startup files is sourced by the window system as it comes up. The user's login shell determines which startup files are sourced, as shown in the following table.

Login Shell	Startup File
C shell	/etc/.login \$HOME/.login
Bourne shell, Korn shell, and all Profile shells	/etc/.profile \$HOME/.profile

Another set of startup files is read whenever a user brings up a shell in a terminal emulator, such as the `cmdtool`, `shelltool`, or `dtterm` (see "Controlling Which Startup Files Are Read When a Shell Comes Up" on page 70).

Controlling the Sourcing of Startup Files

In the Trusted Solaris CDE window system, as in the standard CDE window system, accounts get an editable `$HOME/.dtprofile` file which controls whether the `.login` or `.profile` files are read by the desktop at login. See also the man pages for `login(1)` and `profile(4)`. See “The Sourcing of Startup Files for the Profile Shell User” on page 70 for the one exception to this behavior.

.dtprofile Files

In the Trusted Solaris environment, by default the `.login` or `.profile` files are not sourced by the window system. A `.dtprofile` file controls the sourcing. One of the following `.dtprofile` files is copied into each account’s `$HOME/.dtprofile`:

- An `/etc/dt/config/sys.dtprofile` file that was created by the site’s Security Administrator role, if the file exists, or
- The default `/usr/dt/config/sys.dtprofile`

The following figure illustrates how `$HOME/.dtprofile` is installed.

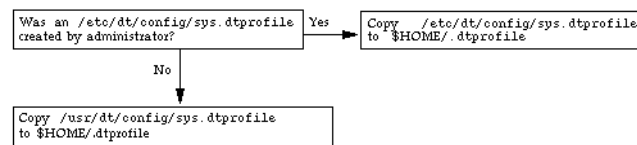


FIGURE 3-1 How `$HOME/.dtprofile` is Installed

In the default `/usr/dt/config/sys.dtprofile`, the `DTSOURCEPROFILE` variable that enables the sourcing of either file is commented out. Removing the `#` before the `DTSOURCEPROFILE` definition in any of the versions of the `sys.dtprofile` file causes the appropriate startup file to be read by the window system.

See the comments in the `/etc/dt/config/sys.dtprofile` file and “To Invoke `.login` or `.profile` During Login” on page 79, if changing the default is consistent with your site’s security policy.

Note – If any modifications to a `.login` or `.profile` accidentally prevent the user from logging in, the user may use the Failsafe Session option on the CDE Login screen. Failsafe Session allows a login without reading any startup files.

The Sourcing of Startup Files for the Profile Shell User

The algorithm for reading `.dtprofile` files when an account has a profile shell as its login shell prevents an account from executing commands that are not permitted to the account.

When a user's login shell is specified as the `pfsh`, `pfssh`, or `pfksh`, the window system does not consult an account's home directory. The window system uses either the default `/usr/dt/config/sys.dtprofile` or a version modified by the Security Administrator role in `/etc/dt/config/sys.dtprofile`.

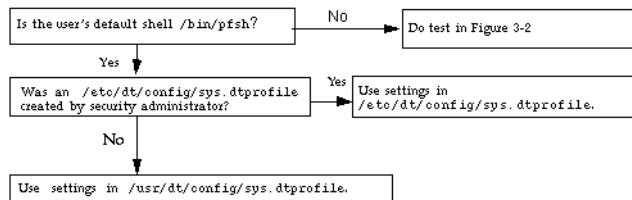


FIGURE 3-2 How `$HOME/.dtprofile` is Bypassed for Profile Shell Users

Controlling Which Startup Files Are Read When a Shell Comes Up

As in the Solaris environment, shell initialization files are used to set search paths and other environment variables and to execute some useful commands and functions. The following table shows which startup files are read by default when each type of shell is launched.

TABLE 3-3 Startup Files Read at Shell Initialization

Shell	Startup File
C shell	<code>\$HOME/.cshrc</code>
	<code>\$HOME/.login</code>

TABLE 3-3 Startup Files Read at Shell Initialization (Continued)

Shell	Startup File
Bourne shell	<code>\$HOME/.profile</code>
Korn shell	<code>\$HOME/.profile</code> file specified with ENV variable
Profile shells (see <code>pfexec(1)</code> man page)	<code>\$HOME/.profile</code>

The `.profile` or `.login` files are invoked only if the shell is also identified as the account's login shell. A shell that is invoked with a prefix of `-` (for example: `- csh`) is a login shell. This means, for example, that when a C shell is started using `csh` (without a `-` prefix), the `.login` file is not executed.

Forcing dtterm to Source `$HOME/.login` or `.profile`

By default, a shell started by `dtterm` is not launched as a login shell. Therefore, the `$HOME/.login` and `$HOME/.profile` files are not read. Any user or role can enable `dtterm` to launch a login shell. See "To Force `dtterm` to Launch New Shells as Login Shells" on page 80 for the procedure.

Note – The default `.profile` file for all roles contains a function to alias the `adminvi(1M)` command to `vi(1)`, but the function does not take effect unless the `Dtterm*LoginShell: true` entry is made in the `$HOME/.Xdefaults-hostname` file. See "To Alias `vi` to `adminvi`" on page 98.

Administering Skeleton Directories

The default skeleton path used by the Users tool in the Solaris Management Console is `/etc/skel`. By default, a set of initialization files for each of the shells are copied from the `/etc/skel` directory into a user account's `$HOME` directory and renamed. The directory `/etc/skel/tsol` exists for role initialization files. The following example shows the default contents of the directory.

EXAMPLE 3-3 Contents of the Default `/etc/skel` Directory

```
trusted% cd /etc/skel
trusted% ls -R
local.cshrc local.login local.profile tsol/
tsol:
role.link_files role.profile
```

In the Trusted Solaris environment, files are automatically copied from the skeleton directory *only* into the SLD at the account's minimum label. Either the user or the administrator must create the files `.copy_files` and `.link_files` as described in "Using `.copy_files` and `.link_files`" on page 72, to ensure that subsequently-created SLDs get copies of initialization files.

Accessing All Man Pages

To ensure that the `man(1)` command can find all of the man pages for the products that combine to make the Trusted Solaris product (Solaris software, CDE, and X windows), the `MANPATH` environment variable should include the directories: `/usr/man`, `/usr/openwin/man`, and `/usr/dt/man`. See "To Set Up Startup Files for Users" on page 77 for an example of adding the path to shell initialization files.

Using `.copy_files` and `.link_files`

The Trusted Solaris files `.copy_files` and `.link_files` are useful for multilevel accounts. The files help to automate the copying or linking of startup files into every SLD in an account's home directory MLD. Whenever a user creates a workspace at a new label, `dtsession` runs the `updatehome(1M)` command to read the contents of `.copy_files` and `.link_files` in the account's minimum label SLD, and copy or link every listed file into the new workspace.

The `.copy_files` file is useful when a user wants a slightly different startup file in different SLDs. Copying is desirable, for example, if users need to use different mail aliases when they are working at different labels. See "To Set Up Startup Files for Users" on page 77 for an example.

The `.link-files` file is useful when a startup file should be identical at any label that it is invoked. For example, if there is one printer to handle labeled print jobs, that printer can be defined once in a startup file, like `.cshrc`. When the `.cshrc` file is listed in the file `.link-files`, `.cshrc` is linked to SLDs at other labels when those SLDs are created. See "To Set Up Startup Files for Users" on page 77 for an example.

The following worksheet provides some examples of common files to copy or link.

TABLE 3-4 Planning Worksheet for .copy_files and .link_files

Common Startup Files	List to be copied (for .copy_files)	List to be copied (for .link_files)
.soffice		
.cshrc		
.dtpfile		
.login		
.Xdefaults		
.Xdefaults- <i>hostname</i>		
.mailrc		
.newsr		
.profile		

Administering cron, at, and batch Jobs

In the Trusted Solaris environment, the Job Scheduler tool in the Solaris Management Console manages jobs. This section describes the difference in managing `cron(1M)` and its associated commands in the Trusted Solaris environment. See the *System Administration Guide, Volume 2* for basic `cron` information. For Trusted Solaris modifications see also the man pages for `at(1)`, `atq(1)`, `atrm(1)`, `cron(1M)`, and `crontab(1)`.

In the default `policy.conf(4)` file, all users are assigned the Basic Solaris User profile. This profile includes the Edit Owned Jobs authorization. The authorization enables users to manage their own `cron` or `at` jobs.

The `crontab` file is generated by a user or role account using the `crontab(1)` command. The `atjob` file is generated by a user or role account using the `at(1)` or `batch` command. In the Trusted Solaris environment, the `crontab`, `at`, and `batch` commands must be in one of the account's rights profiles.

In the Trusted Solaris environment, the `crontabs` and `atjobs` spool directories are MLDs that hold job files at different labels. With MLDs as spool directories, one user can have multiple `crontab` files at different labels within the `crontabs` directory, and, similarly, one user can have multiple `atjob` files at different labels within the `atjobs` directory.

Running a Job with a Profile Shell



Caution – If a job requires that a profile shell execute the job, the Security Administrator role must ensure that all of the job’s commands are also in a rights profile assigned to the invoking user.

`cron` jobs can be executed using a profile shell. Profile shells are documented on the `pfexec(1)` man page. A profile shell can execute a `cron` job if:

- The invoking account’s login shell is the one of the profile shells or
 - The `$SHELL` environment variable is set to `/bin/[pfsh|pksh|pcsh]`
- Otherwise, the `cron` program uses the default Bourne shell, `sh(1)`, for `cron` jobs.

For `at` jobs there is a third case in which the profile shell is used. A user can use the `at` program with the `-c` (for `csh`), `-k` (for `ksh`), `-s` (for `sh`), option along with the `-P` (for profile shell) option to specify the shell which should run the job. Therefore, `at` jobs are executed in the profile shell if:

- The invoking account’s login shell is one of the profile shells or
 - The `$SHELL` environment variable is set to a profile shell or
 - The `at` command is specified with the `-P` option
- If none of the previously described conditions apply, the `at` program uses:
- Any shell specified with either the `-c`, `-k`, or `-s` options or
 - The default shell, `sh`

Running Privileged Commands in Scheduled Jobs

If a command in an `at` or `cron` job needs to run with privileges, either forced or inheritable privileges may be made available. Enabling a command to run with forced privileges, so that the privileges apply no matter who executes the command, is insecure practice. Therefore, the Security Administrator role typically does the following to make the privileges available by inheritance:

1. Specify the command and any privileges it needs in one of the invoking user’s profiles using the Rights tool in the SMC.
2. Specify that the job is executed with a profile shell, as described in “Running a Job with a Profile Shell” on page 74.

For more information, see “Assigning Inheritable Privileges to a Command or Action” on page 230.

For a `cron` job example, see “To Write a Profile Shell Script” on page 249.

How the UNIX Domain Socket is Used for Communications

The communication mechanism between `crontab(1)`, `at(1)`, `atrm(1)`, and `cron(1M)` in the Trusted Solaris environment is a UNIX domain socket. See the man page for `libt6(3NSL)`.

The `cron(1M)` command is modified to create and bind the UNIX domain socket at `ADMIN_LOW` to `/etc/cron.d/CRON`. The `/etc/cron.d/CRON` file is also used as a lock file to prevent more than one execution of the clock daemon.

After it creates the UNIX domain socket, the `cron` command is changed to run at `ADMIN_HIGH`. The `/var/cron/log` file is created by the clock daemon at `ADMIN_HIGH`. The clock daemon logs its internal messages in this log file.

An ancillary file is created in the `crontabs` MLD for each `crontab` file and in the `atjobs` MLD for each `atjob` file. Modification of a `crontab` or an `atjob` file also changes the ancillary file data. The ancillary file is named `username.ad` for a `crontab` file, and `jobname.ad` for an `atjob` file. The ancillary file contains information used by `cron` to set up a job.

Trusted Solaris software is delivered with the following `/var/spool/cron/crontabs` files:

- At the `ADMIN_LOW` label, pairs of `crontab` and ancillary files for `root`, `uucp`, `adm`, and `sys`.
- At the `ADMIN_HIGH` label, pairs of `crontab` and ancillary files for `root` and `lp`.

Permitting Users to Access Others' Jobs

The default Trusted Solaris security policy does not permit users to access jobs owned by other users. To enable certain users to access jobs belonging to other users, the Security Administrator role can edit system files, or assign to the user a network-wide authorization through the Scheduled Jobs tool in the Solaris Management Console.

Conditions for Access to Other's Jobs

An account invoking the `at`, `atq`, `atrm`, or `crontab` commands can look at, edit, or remove jobs belonging to another user only when the following conditions are met.

For the `at`, `atq`, or `atrm` commands, the following conditions must apply:

1. The specified `username` or the `username` of the specified job's owner is one of the special system account names listed in the `at.admin` file and condition 3 is true,

or

2. The *username* of the specified at job's owner is the name of a role account and condition 3 is true.
3. The account has the Edit Owned Jobs authorization in a rights profile.
4. If neither condition 1 nor condition 2 is true, the invoking account must have the Manage All Jobs authorization in a rights profile.

For the `crontab` command, the following conditions must apply:

1. The specified *username* is one of the special system account names listed in the `cron.admin` file and condition 3 is true, or
2. The specified *username* is one of the role account names and condition 3 is true.
3. The invoking account has the Edit Owned Jobs authorization.
4. If neither of 1 or 2 is true, the invoking account must have the Manage All Jobs authorization in a rights profile.

Assigning the SMC to Normal User Accounts

The Basic Solaris User profile provides the authorizations to view most information in the Solaris Management Console (SMC) tools. The default `/etc/security/policy.conf` file ships with an entry that assigns the Basic Solaris User profile to all users.

If the administrator has removed the Basic Solaris User rights profile from the `policy.conf(4)` file or assigns this rights profile only to selected users, most users will not be able to view the SMC information.

By default, only administrative roles have the required authorizations in their rights profiles to modify information in the SMC. While a Security Administrator can assign one or more of the required authorizations to a user account, such assignment bypasses the trusted path.

Preparing for User Accounts (Tasks)

▼ To Modify Default User Label Attributes

When these changes are made while the system is being configured on the name service master, the files can be copied to each client computer as it is configured.

1. **Assume the Security Administrator role and go to an ADMIN_HIGH workspace.**
2. **Review the default user attribute settings in the `/etc/security/tsol/label_encodings` file as shown in Table 3–1.**
3. **Modify the user security attributes in the `label_encodings` file by using the Edit Label Encodings action from the System_Admin folder.**
The `label_encodings` file should be the same on all hosts.
4. **Distribute a copy of the file to every Trusted Solaris host.**
Follow the procedure in “(Optional) Copy Network Files to the `/etc` Directory” in *Trusted Solaris Installation and Configuration*.

▼ To Modify `policy.conf` Defaults

When these changes are on the name service master, every user and role that is created inherits these values.

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace. If you are using a name service, do this on the name service master.**
2. **Review the default settings in the `/etc/security/policy.conf` file as shown in Table 3–2.**
3. **Modify the settings in the `policy.conf` file by opening it in the Admin Editor and saving the new settings.**

▼ To Set Up Startup Files for Users

Users can put a `.copy_files` or `.link_files` into their home directory MLD at the SLD that corresponds to the minimum sensitivity label or can modify the files in the

minimum label SLD if the files are already there. This procedure is for the administrator role to automate the setup for a site.

1. **Assume the System Administrator role and go to an ADMIN_LOW workspace. If you are using a name service, do this on the name service master.**

2. **Set appropriate defaults for users in shell initialization files.**

For example, set the MANPATH to enable the users to access all Trusted Solaris man pages, and the LPDEST and PRINTER environment variables to send print jobs to a labeled printer.

In a startup .cshrc file for users:

```
setenv MANPATH /usr/dt/man:/usr/openwin/man:/usr/man
setenv PRINTER conf-label-gluten
setenv LPDEST conf-label-gluten
```

In a startup .ksh file for users:

```
export MANPATH /usr/dt/man:/usr/openwin/man:/usr/man
export PRINTER conf-label-gluten
export LPDEST conf-label-gluten
```

3. **Go to your directory of startup files for users.**

For example,

```
$ cd ~/startfiles_Cusers
```

4. **Copy the generic copies of startup files into a skeleton directory.**

The following example adds startup files for the mailer, the C shell, dtlogin, and dtterm.

```
$ cp .cshrc .login .mailrc .Xdefaults .Xdefaults-hostname \
/etc/skelC
```

5. **Open the Admin Editor from the System_Admin folder in the Application Manager and enter the pathname to the .copy_files file, /etc/skelX/.copy_files.**

In the example, the pathname is /etc/skelC/.copy_files.

6. **Type into .copy_files, one file per line, the files to be copied into the user's home directory at all SLDs. Save and quit the file.**

Use Table 3–4 for ideas. For example,

```
# .copy_files for regular users
# Copy these files to all home directory SLDs
.mailrc
.netscape
.soffice
:wq
```

7. **Repeat for `.link_files` — create the file and type into it the files to be linked from the user's minimum-label home directory SLD to higher SLDs. Save and quit the file.**

Use Table 3–4 for ideas. In step 4 the administrator placed site copies in the `/etc/skelC` directory. For example,

```
# .link_files for regular users with C shells
# Link these files to all home directory SLDs
.cshrc
.login
.Xdefaults
.Xdefaults-hostname
:wq
```

▼ To Invoke `.login` or `.profile` During Login

Note – This procedure changes the default for all users on the host where the change is made.

1. **Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.**
2. **Use the File Manager or the command line to copy `sys.dtprofile` from `/usr/dt/config` to `/etc/dt/config`.**
Create the destination directory if it does not already exist.

```
$ mkdir /etc/dt/config
$ cd /usr/dt/config
$ cp sys.dtprofile /etc/dt/config
```
3. **Use the Admin Editor action in the `System_Admin` folder to open the `sys.dtprofile` file for editing.**
See “To Edit a Local File” on page 33, if needed.
4. **Remove the pound sign (#) before the `DTSOURCEPROFILE` variable assignment at the end of the file.**
After editing, the line should look like the following:

```
DTSOURCEPROFILE=true
```
5. **Save and close the file.**
6. **Distribute a copy of the file to every Trusted Solaris host that should have this change.**
Follow the procedure in “(Optional) Copy Network Files to the `/etc` Directory” in *Trusted Solaris Installation and Configuration*.

▼ To Force dtterm to Launch New Shells as Login Shells

Do this procedure once for each home directory SLD at which the account works, or do it once in the home directory SLD at the account's minimum label, and then list the `.Xdefaults-hostname` in either `.copy_files` or `.link_files`, as described in "Using `.copy_files` and `.link_files`" on page 72.

1. Go to your home directory.
2. Use a text editor to create or modify the `.Xdefaults-hostname` file.
3. Add the following entry.

```
Dtterm*LoginShell: true
```
4. Write and quit the file.
5. Log out and log back in to put the change into effect.

▼ To Customize Shell Initialization Files for Users

1. In the System Administrator role, follow the procedure to customize user initialization files in the section "Setting Up User Accounts Task Map" in *System Administration Guide, Volume 1*.
The procedure tells you how to create three shell-specific skeleton directory names that are entered into the Skeleton path field when creating a user. The procedure also tells you to copy the `local.login` file to the `skelC` subdirectory, the `local.profile` file to the `skelK` subdirectory and the `local.login` file to the `skelB` subdirectory.
2. Create a `skelP` subdirectory for users whose default shell is a profile shell.
3. Create a `.profile` file and copy it into the `skelP` directory.
See "Customizing a User's Work Environment" in *System Administration Guide, Volume 1* for a discussion of what to include in initialization files.
4. When creating user accounts or user templates, enter the appropriate `skelX` pathname into the Skeleton path field.

▼ To Enable a User to Track Others' Jobs on a System

A Security Administrator might want to set up a server, such as an audit administration server, so that the main user of the server can monitor all at jobs and cron jobs running on the system.

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**
2. **Open the Admin Editor from the System_Admin folder and type in the filename /etc/cron.d/cron.admin.**
3. **Add the user's username to the list.**
The user is now enabled to track others' cron jobs on this system.
4. **Repeat for the file /etc/cron.d/at.admin, if the user is permitted to track other users' at jobs on this system.**
5. **Check that the user has the Basic Solaris User profile.**
The Basic Solaris User profile includes the Edit Owned Jobs authorization.

▼ To Enable a User to Track All Others' Jobs

The ability to monitor the cron and at jobs of other users is typically restricted to a role. However, the Security Administrator can assign an authorizations to enable a user to monitor others' jobs.

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**
2. **Create a help file for the new right that you are about to create. Name it RtManageJobs.html, and provide it with text.**
For example,

```
<HTML>
<BODY>
<p>
This right allows users to manage other users' cronjobs and atjobs.
The user can modify cronjobs and atjobs using the Jobs Scheduler.
</BODY>
</HTML>
```

See "To Create a Help File for a Rights Profile" on page 89 for details of the steps.
3. **Launch the Solaris Management Console, and choose a toolbox in the appropriate scope. If you are using a name service, open the toolbox in the name service scope.**
4. **Click the Users tool and supply a password when prompted.**

5. **Double-click the Rights tool, and create a Custom_Manage_Jobs rights profile that contains the Manage All Jobs authorization.**

The Manage All Jobs authorization allows the account to add, modify or delete any user's job and to modify cron policies in the Job Scheduler tool of the SMC.

See "To Create a Rights Profile" on page 89 for the steps. Substitute the names and help text for your new profile in the procedure.

6. **Assign the right to the user by following the steps in "To Assign an Authorization to a User" on page 93.**

Managing Users and Rights With SMC

This chapter describes how to add rights and users. It also points out system and security requirements when configuring users. This chapter includes the following tasks:

- “To List All Rights” on page 88
- “To Create a Help File for a Rights Profile” on page 89
- “To Create a Rights Profile” on page 89
- “To Modify a Rights Profile” on page 90
- “To Create a User Template” on page 91
- “To Add a User Account” on page 92
- “To Modify a User Account” on page 92
- “To Assign a Right to a User” on page 93
- “To Assign an Authorization to a User” on page 93

Before Setting Up User Accounts

If you are using a name service, you *must have* the following set up before creating accounts:

- The home directory server must be mounted on the name service master.
- The mail server must be mounted on the name service master.
- The `tsol_smc.tbx` and `tsol_name-service.tbx` toolboxes must be edited with the name service master's name and address on the name service master. The install team set this up in “Edit SMC Toolbox Definition for the Name Service” in *Trusted Solaris Installation and Configuration*.

The following information is useful to have when you set up users in the SMC Users tool.

- A list of the available rights profiles.

The “Rights Profile Descriptions” in *Trusted Solaris Administration Overview* lists the rights profiles that are delivered with the Trusted Solaris release. If additional rights profiles have been created at your site, see your own internal documentation for a description. See “To List All Rights” on page 88 for how to list the rights profiles on your system and in your name service. The `smprofile(1M)` man page also provides examples.
- A list of the available roles.

See “To List All Roles” on page 100 for how to list the roles in your name service. The `smrole(1M)` command with the `list` option lists all the roles. If your site has added or modified roles, see your internal documentation for a description of the site’s roles.
- A list with the name of each person who needs an account, along with the responsibilities, roles, clearances and minimum labels assigned to each. You also need to determine who is going to be able to assume administrative roles.

Adding or Modifying a User Account

The System Administrator role creates user accounts. The Security Administrator role sets up the security aspects of an account, for example, the account’s password. The roles use the Solaris Management Console (SMC) 2.0, a GUI-based “umbrella application” that serves as the launching point for administrative tools. The User Accounts tool in the SMC provides two ways to create a user — from scratch using a “wizard”, or from a template that you create.

Using the Wizard requires no preparation. However, its defaults cannot be changed and may not be appropriate for your site. For example, the default login shell is the Bourne shell, and the home directory server is the local system.

Creating one or more User Templates enables you to set a reasonable set of defaults for all new user accounts. Once the template is created, its defaults are added to the default user security attributes that are defined outside the SMC. The template plus the security defaults can create users whose attributes suit your users’ preferences and your site’s security policy.

Properly configured system-wide settings and one or more user templates enable the System Administrator role alone to add users. The Security Administrator role then can assign user security attributes, such as a password and a role assignment, when the system is ready for users to log in. See “To Create a User Template” on page 91 for the procedure to create a template.

The following table shows the default values provided by the Wizard, and additional fields that can be set in a template. Note that you must specify or change values in the Wizard for each user you create, whereas you specify a value in a Template once. The Template value is then applied to every user that is created with that template.

TABLE 4-1 Configurable User Attributes in a Wizard versus a Template

User Attribute	Wizard Value	Possible Template Value
Login Shell	Bourne	Bourne, Korn, C, profile, others
Account Availability	Always Available	Always Available, Locked, or Available until <i>date</i>
Primary Group	specify any existing group	specify any existing group
Additional Groups	none — cannot be set	specify any existing group
Home Directory Path	specify path	specify path
Home Directory Server	specify server	specify server
Home Directory Sharing	-rwxr-x—x	specify permissions on home directory
Copy Initial Files From	/etc/skel	specify a skeleton directory
Automount server?	automounted	specify whether to automount
Mail Server	specify	specify

As shown in the above table, a template will simplify user creation if your site includes users whose default shell is not Bourne, whose account should expire or be locked, who should belong to several groups, whose home directory and mail server are not on the local system, or who has skeleton files in a directory other than `/etc/skel`.

If a shell-specific subdirectory has been created for each of the shells, you need to enter the correct `skelX` subdirectory name into the Skeleton path field in the Copy Initial File From: field in a template.

Security information must still be entered by the Security Administrator, as the following table shows. For information about the files that contain default values, see “Managing Default User Security Attributes” on page 65.

TABLE 4-2 User Security Attributes Assigned after Creation

User Attribute	Source of Default Value
User Password	none — must be assigned by Security Administrator
Rights	<code>policy.conf</code> file

TABLE 4-2 User Security Attributes Assigned after Creation (Continued)

User Attribute	Source of Default Value
Roles	none — must be assigned by Security Administrator
Labels	label_encodings file
Visible Labels	policy.conf file
Account Usage	policy.conf file
Audit	none — must be entered by Security Administrator

Assigning Passwords to Users

The Security Administrator role assigns passwords to users after the user has been created. The password can be generated (Choose from List) or created by the administrator. The Security Administrator also specifies whether the user can pick a new password, or must choose one from a generated list when changing passwords.

Unlike the Solaris environment, the Trusted Solaris environment requires users and roles to use the TP (Trusted Path) menu Change Password option to change their own passwords. They do not use the command line.

Users can be forced to change their passwords at regular intervals. The password aging options limit how long any intruder who is able to guess or steal a password could potentially access the system. Establishing a minimum length of time to elapse before change also prevents a user with a new password from reverting immediately to the old password.

Note – The passwords for users who can assume roles should not be subject to any password aging restraints.

Assigning Rights to Users

The Security Administrator role assigns rights to users after the user has been created. The administrator may assign no rights profiles (by default users get the Basic Solaris User rights profile), one rights profile, or multiple rights profiles.

The order of profiles is important. When the user invokes a command or action, the profile mechanism uses the first instance of the command or action in the account's profile set, with whatever attributes have been defined for the command or action in the profile where it is found.

You can use the sorting order of profiles to your advantage. If you want a command to run with different privileges from those defined for it in an existing profile, create a new profile with the desired privilege assignments for the command and insert that new profile so that the profile mechanism finds the new one first.

Note – Do not assign any role profiles to a normal user account. Doing so would introduce some measure of risk and a good deal of confusion. Many of the administrative role's commands and applications do not work outside of the administrative role workspace because they require the trusted path attribute.

Assigning Roles to Users

The Security Administrator role assigns roles to users after the user has been created. A user is not required to have a role. A single user can have one or more roles if having more than one role is consistent with your site's security policy.

Assigning Trusted Solaris Attributes to Users

The Security Administrator role assigns Trusted Solaris Attributes to users after the user is created. If the administrator has set up correct defaults, assigning Trusted Solaris attributes is needed only for users who are exceptions to the defaults.

Assigning Audit Classes to Users

The Security Administrator role can assign audit classes to users after the user account is created. These audit classes are exceptions to the audit classes set up in the `/etc/security/audit_control` file on the system. See *Trusted Solaris Audit Administration* for more about auditing.

Adding or Modifying a Rights Profile

The Security Administrator role creates new rights profiles and modifies existing ones. A right or rights profile is a collection of actions and commands with security attributes, plus authorizations. A rights profile can be part of another rights profiles, or stand on its own.

The Rights tool in the SMC GUI displays the existing rights profiles. Each rights profile has a corresponding help file that displays in the GUI. The administrator should create a right's accompanying help file before creating the right itself. The GUI prompts for the name of the help file when the administrator creates the new right. See "To Create a Help File for a Rights Profile" on page 89 and "To Create a Rights Profile" on page 89.

Managing Users and Rights (Tasks)

▼ To List All Rights

Once the SMC is initialized, users or roles can view one rights profile at a time in the Rights tool under Users in the SMC, or use the `smprofile(1M)` command described below to see a list of all profiles.

1. Assume the Security or System Administrator role.
2. To list the rights profiles in a name service domain, use the `smprofile list` command with the `-D` option to specify the *name_service_type:server_name/domain_name*. Provide a password when prompted.

The following example lists the profiles that are defined in the NIS+ domain `tropics.example.com` whose NIS+ master server is `toucan`. The command is being executed on the `tern` system:

```
$ /usr/sadm/bin/smprofile list -D nisplus:/toucan/tropics.example.com —
Authenticating as user: janez
```

```
Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: rolePassword
Loading Tool: usermgr.cli.profile.UserMgrProfCli from tern
Login to tern as user janez, role admin was successful.
Download of usermgr.cli.profile.UserMgrProfCli from tern was successful.
Profile name: All Actions Description: A complete set of actions
(no commands) without any privilege.
Profile name: All Authorizations Description: Grant all authorizations.
Profile name: All Commands Description: A complete set of commands
(but no actions) without any privilege.
Profile name: All Description: Execute all commands and actions.
...
Profile name: User Security Description: Manage passwords, clearances.
Profile name: Trusted Edit Description: Use the trusted_edit script
when editing.
```

The following example lists the security attributes of the All profile.


```
$ /usr/sadm/bin/smpprofile list \
-D nisplus:/toucan/tropics.example.com -- -l -n All
...
Profile name:    All
Description:     Execute all commands and actions
help:           RtAll.html
Command:        *;*;*;*;*
  policy:       tsol
  type:         act
Command:        *
  policy:       tsol
  type:         cmd
```

▼ To Create a Help File for a Rights Profile

1. Assume the Security Administrator role and go to an **ADMIN_LOW** workspace.
2. Using the Admin Editor, open a new help file with an **.html** extension in the directory **/usr/lib/help/profiles/locale/locale/**.
For example, **FilePriv.html**.

3. In the help file, describe the rights profile you have added.

Follow the online help when creating the help file. For example:

```
<HTML>
<HEAD>
Copyright (c) 2001 by Sun Microsystems, Inc. All rights reserved.
<!-- SCCS keyword #pragma ident    "%Z%M% %I%    %E% SMI; TSOL 2.x" -->
<!-- FilePriv.html -->
</HEAD>
<BODY>
Allows a user to specify the allowed and forced privileges to be associated
with the execution of a program file.
<P>
If the name of the file privileges rights profile is grayed, you are not
allowed to add or remove it.
</BODY>
</HTML>
```

4. Save and quit the new help file.
5. Enter the name of the help file in the **Help File Name:** field when creating the right.

▼ To Create a Rights Profile

1. Assume the Security Administrator role and go to an **ADMIN_LOW** workspace.

2. Create a help file for the new rights profile.

Use the procedure “To Create a Help File for a Rights Profile” on page 89.

3. Bring up the SMC in the desired scope and click the Users tool. Supply a password when prompted.

4. Double-click the Rights tool.

5. To create a rights profile, select Add Right from the Action menu.

Use the online help when creating the new right.

6. Name the profile Custom *rolename* Role, and describe it.

For example, for a role whose username is `auditadmin`, you would create an empty Custom Auditadmin Role profile. In the profile’s description you would enter:

```
Modify this rights profile to customize the Audit Administrator role
```

7. Select the action or command to add to the right.

See the Trusted Solaris man pages for individual commands for the security attributes needed by the command or any of its options to succeed. For example, if the command requires privilege to accomplish a task, adding the privilege to the command enables it to execute with the specified inherited privileges when a user or role has been assigned this rights profile.

8. Click the Set Security Attributes button to enter the information requested in the help for the Ownership and Extended Attributes areas.

For example, by adding the name of an installation program to a rights profile, assigning to the program a real UID of 0, and then assigning the profile to a role, the Security Administrator can enable an installation program to succeed when run by a role that has another UID, such as the System Administrator role.

9. Add authorizations if needed.

A rights profile can contain commands only, actions only, authorizations only, or a combination of commands, actions, and authorizations.

10. Click OK to save the new rights profile.

▼ To Modify a Rights Profile

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.

2. Bring up the SMC in the desired scope and click the Users tool. Supply a password when prompted.

3. Double-click the **Rights** tool, select the profile, then choose **Properties** from the **Action** menu.

Refer to the online help when modifying the right.

4. Click **OK** to save the rights profile.

▼ To Create a User Template

1. Assume the **System Administrator** role and go to an **ADMIN_LOW** workspace.
2. Bring up the **SMC Users** tool. If you are using a name service, do this procedure on the name service master.
3. Supply a password when prompted.
4. Click the **User Templates** tool.
5. Choose **Add User Template** from the **Action** menu.

6. **Decide on a Login Shell.**

You can assign a profile shell to users by choosing **other** and then typing in the path to a profile shell: `/bin/pfsh`, `/bin/pfcsch`, or `/bin/pfksch`. While working in a profile shell, an account can execute only those commands that are in the account's set of profiles. See the `pfexec(1)` man page for descriptions of the profile shells.

The Bourne, Korn, and C shells allow the account to execute all available commands that do not need to inherit privilege. See the following man pages for more information about the listed shells: `csh(1)`, `ksh(1)`, `bash(1)`, `tsh(1)`, `zsh(1)`.

7. **Use the online help to guide you through the General, Group, Home Directory, Sharing, Password Options, and Mail tabs.**

The following is a text example of a User Template.

```
Template Name: Desktop User
Template Description: Users with C-shells
General - Login shell = C Shell
          - Account is Always Available
Group    - Primary Group = staff
          - Secondary Groups = writers
Home Directory - Server = /net/egret.aviary
               - Path   = /net/egret/export/home
                 Append User Names to path above
               - Skeleton directory = /etc/skel/Csh
               - Automatically mount
Home Directory Sharing
          - Group members have Read-only Access
          - All users have Read-only Access
Password Options - User must keep for 31 days
                  - Before change, alert user 5 days
```

```
- User must change within 5 days
- Expires if not used for 31 days
Mail Server - /net/pigeon.aviary
```

8. Click OK when finished to save the template.

▼ To Add a User Account

1. Assume the System Administrator role and go to an ADMIN_LOW workspace.
2. Bring up the SMC in the desired scope and click the Users tool. Supply a password when prompted.
3. Double-click User Accounts.

All configured users are displayed as icons labeled with their usernames.

4. Choose one of the following from the Action menu:

- Add User->With Wizard
- Add User->From Template

To use the From Template option, you need to first create a template. See “To Create a User Template” on page 91 for the procedure.

Depending on whether you use the Wizard or Template method, some fields will not be available.

5. Enter the user’s name and ID.

User names and UIDs must be unique to ensure traceability of activities back to a single identified user. Therefore, each user name and UID should not be duplicated anywhere on the network, and should not be reused during the life of the system.

6. Enter a description.

The description will appear in the user’s email From field. For example,

```
From: Bar Bar -- Useful Worker
```

7. Continue to create the user, referring to the online help when necessary.

Also see “Adding or Modifying a User Account” on page 84 for guidance.

▼ To Modify a User Account

The Security Administrator role follows this procedure to add user security attributes, such as passwords, to user accounts.

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.

2. Bring up the SMC in the desired scope and click the Users tool. Supply a password when prompted.
3. Double-click the User Accounts tool, highlight the username, then choose Properties from the Action menu.
Use the online help when modifying the user's properties.
4. Click OK when you have entered all the changes.

▼ To Assign a Right to a User

The right must exist before assigning it. See "Managing Users and Rights (Tasks)" on page 88 for creating a rights profile.

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
2. Bring up the SMC in the desired scope and click the Users tool. Supply a password when prompted.
3. Double-click the User Accounts tool, highlight the username, then choose Properties from the Action menu.
Use the online help when modifying the user's security attributes.
4. Click the Rights tab.
5. Order the rights profiles appropriately. Move the new right above the All profile, and further up if necessary.
When the user invokes a command or action, the profile mechanism uses the first instance of the command or action, with its security attributes. So, if two rights profiles share a command, and both are assigned to one user or role, the command in the first profile is executed, with its attendant security attributes. The same command in the second profile is not seen by the profile mechanism.
6. Click OK when you have entered all the changes.

▼ To Assign an Authorization to a User

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
2. Launch the Solaris Management Console and choose the appropriate scope. If you are using a name service, choose the name service scope.
3. Click Users and supply a password when prompted.

4. **Double-click the Rights tool to create a new right with the authorization, if necessary. Save the new right.**
See “To Create a Help File for a Rights Profile” on page 89 and “To Create a Rights Profile” on page 89 for the steps.
5. **Add the new rights profile to the user’s rights by opening the User Accounts tool, selecting the user, and editing the user’s properties.**
Use the online help for guidance.

Managing Roles

This chapter provides background about administrative roles and describes how to expand a role's powers. This chapter contains the following procedures:

- "To Alias vi to adminvi" on page 98
- "To Assign the trusted_edit Editor to a Role" on page 99
- "To Alias vi to trusted_edit" on page 100
- "To List All Roles" on page 100
- "To Modify a Role" on page 101
- "To Configure a New Role" on page 102
- "To Enable a Role to Administer NIS+" on page 102
- "To Enable Remote Role Assumption from Untrusted Systems" on page 103

Roles and the Trusted Path Attribute

Roles in the Trusted Solaris environment divide the privileges of the `root` account into discrete accounts. In the Trusted Solaris environment, privileged actions require the *trusted path attribute*.

The trusted path attribute is available during boot and when a program is launched from a role workspace.

Many administrative commands and actions require the trusted path attribute. For instance, when the Solaris Management Console (SMC) is launched on behalf of an administrative role, the SMC checks for the trusted path attribute.

Allowing Remote Logins by Administrative Roles

In both the Trusted Solaris and Solaris environments, remote logins may or may not require authorization. “Managing Remote Logins” on page 67 describes the conditions and types of logins that require authorization. The administrative roles by default have the Remote Login authorization, but the Security Administrator role needs to change a setting in the `/etc/default/login` file on each computer where administrative roles work, to allow remote logins from that computer. See “To Enable Any Role to Log In Remotely” on page 35 for the procedure.

Creating a New Role

Sites may create a new administrative role to enable a users to perform a defined set of administrative tasks. While in the role, the users share the role’s home directories (at different labels) and ownership of files. A site might create a new administrative role, for example, to handle auditing review.

If the new role needs capabilities that the Security Administrator role does not have to give away, the Primary Administrator role creates the new role. The procedure is described in “To Modify a Role” on page 101.



Caution – If site security policy permits, root’s capabilities can be extended to allow root to do NIS+ administration from a NIS+ client, although this is not recommended. See “To Enable a Role to Administer NIS+” on page 102 for the procedure.

A new role may require a new profile. If the profile needs capabilities that the Security Administrator role does not have, the primary administrator role must create the profile.

Before creating the profile, the Security Administrator role should analyze whether any of the commands or actions in the new profile need privilege to be successful, as described in “To Find Out Which Privileges a Program Needs” on page 252. See the man pages for individual commands for the *required* and *override* privileges a command might need.

Modifying a Role With the SMC

Adding or modifying a role account in the SMC is similar to adding or modifying a user, with the following exceptions.

- **The Security Administrator role** – Creates roles. There is no Wizard or Template for adding a role.
- **Administrative Roles** – The Security Administrator role uses the SMC Administrative Roles icon to create roles.
- **Role Mailing List** – By default, a role mailing list is created.
- **Login Shell** – A role must have a profile shell for its login shell. In the SMC GUI, a profile shell is called an “Administrator’s Shell”. While working in a profile shell, the role can execute only those commands that are in its set of rights profiles. See the `pfexec(1)` man page for descriptions of the profile shells.
- **Group** – Each role becomes a member of the `sysadmin` group 14 by default.
- **Rights** – Except for the root role, which is shipped with a set of rights already assigned, each of the recommended roles has a predefined rights profile. Creating the roles by assigning the appropriate rights profiles is described in “Creating Roles and Users” in *Trusted Solaris Installation and Configuration*.
- **Label View** – By default, the label view is Internal, not External. Roles do not use the `/etc/security/policy.conf` file for default values.

Customizing Profiles for the Recommended Roles

Custom role profiles are used when customizing default profiles that are assigned to the recommended roles. Making all changes for each role in its custom profile makes it much easier to debug problems or characterize the system if you ever need to call for service.

- The root role has the Custom root Role profile assigned by default.
- The Custom `secadmin` Role profile is nested in the Rights Security profile, which is assigned to the Security Administrator during configuration, as described in “Creating Roles and Users” in *Trusted Solaris Installation and Configuration*.
- Before creating a new role or modifying the Security Administrator, Primary Administrator, or Operator roles, you need to create a Custom *rolename* Role and then assign it to the role.

See also “To Configure a New Role” on page 102.

- Before modifying the profiles for any role, make sure the appropriate Custom *rolename* Role is assigned, and make the changes in the custom profile.

Enabling Role Assumption from Untrusted Systems

In the Trusted Solaris environment, users assume roles through the Trusted Path menu. The roles then operate in protected trusted workspaces. By default, roles cannot be assumed outside of the trusted path. If site policy permits users on unlabeled hosts that are running SMC 2.0 client software, to assume a role and administer trusted hosts, the Security Administrator role can change the default policy.

See “To Enable Remote Role Assumption from Untrusted Systems” on page 103 for the procedure.

If this policy change is made, it only applies when the user on the remote untrusted computer has an account on the Trusted Solaris system with the ability to assume the desired role.

Managing Roles (Tasks)

▼ To Alias vi to adminvi

By default, all roles have the `adminvi(1M)` editor and the `dtterm` terminal assigned by default.

The default `profile(4)` file in the home directories for all roles has the following function to alias `vi` to `adminvi`:

```
vi() { adminvi $1 ; }
```

However, the alias does not by defaults since the `.profile` file is not sourced by default. For more information about startup files in the Trusted Solaris environment, see the discussion in Chapter 3 under “Managing Initialization Files” on page 67.

1. Assume the System Administrator role and go to an **ADMIN_LOW** workspace.
2. Create an **.Xdefaults** file for each hostname the role uses.
For example, when the role works on computers named **tern** and **toucan**, create a **\$HOME/.Xdefaults-toucan** and a **\$HOME/.Xdefaults-tern**.
3. Add the following entry to the **\$HOME/.Xdefaults-*hostname*** file in the SLD at each label at which the role works:

```
Dtterm*LoginShell: true
```
4. Make sure that a copy of the file is in every SLD at which the role works.
See “Using **.copy_files** and **.link_files**” on page 72 for how to copy or link files into every SLD.

▼ To Assign the **trusted_edit** Editor to a Role

The **/usr/dt/bin/trusted_edit** script is a wrapper that launches an editing window using the **\$EDITOR** environment variable. The wrapper audits all changes.

1. Assume the Security Administrator role and go to an **ADMIN_LOW** workspace.
2. If the **trusted_edit** command is not in one of the role’s profiles, use the **SMC Rights** tool to add the **trusted_edit** command to the Custom *rolename* profile.
Refer to the online help when modifying the rights profile.
 - a. Add the **/usr/dt/bin/trusted_edit** script to the Custom *rolename* profile.
 - b. Give the script the **proc_audit_tcb** privilege.
3. Make sure that the role has the Custom *rolename* profile assigned to it or subsumed in one of its assigned rights.

▼ To Alias vi to trusted_edit



Caution – Because `trusted_edit` launches a window, it cannot be used for command line editing. Command line editing may be the only option available in a remote login session, so for this reason, do not assign `trusted_edit` as the only editor for a role, unless the role never needs to do remote editing on the command line.

1. Search for the `vi` function in the `.profile` file in the role's home directory:

```
vi() {adminvi $1;}
```

2. Replace `adminvi` with `trusted_edit`:

```
vi() {trusted_edit $1;}
```

3. Make sure the following entry is also made in the `$HOME/.Xdefaults-hostname` file in the SLD at each label at which the role works:

```
Dtterm*LoginShell: true
```

Note – Make a file for each hostname the role uses. For example, when the role works on computers named `tern` and `toucan`, create a `$HOME/.Xdefaults-toucan` and a `$HOME/.Xdefaults-tern` in each SLD.

▼ To List All Roles

Once the SMC is initialized, users or roles can use the `smrole(1M)` command described below to see a list of all roles.

1. Assume the Security or System Administrator role.
2. To list the roles in a name service domain, use the `smrole list` command with the `-D` option to specify the `name_service_type:server_name/domain_name`. Provide a password when prompted.

The following screen example lists the roles that are defined in the NIS+ domain `tropics.example.com` whose NIS+ master server is `toucan`. The command is being executed on the `tern` system:

```
$ /usr/sadm/bin/smrole list -D nisplus:/toucan/tropics.example.com --
Authenticating as user: admin
```

```
Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: rolePassword
Loading Tool: usermgr.cli.role.UserMgrRoleCli from tern
Login to tern as user janez, role admin was successful.
```

```
Download of usermgr.cli.role.UserMgrRoleCli from tern was successful.
admin          100          System Admin
secadmin        101          Security Admin
oper           102          Operator
primaryadmin    104          Primary Administrator Role
root            0           Super-User
```

3. To list the roles defined in the local system, use the `smrole list` command followed by the double dash `--`.

```
$ /usr/sadm/bin/smrole list --
```

To list all roles defined in local files on a system named `tern`:

```
/usr/sadm/bin/smrole list -- -h tern
```

▼ To Modify a Role

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.

If the role requires authorizations that the Security Administrator cannot grant, the Primary Administrator role must change the role.

2. Bring up the SMC in the desired scope and click the Users tool. Supply a password when prompted.

3. Double-click the Rights tool, then select the Custom *rolename* Role.

To modify a role, you modify the contents of one or more of its rights profiles.

4. Make any needed modifications to the Custom *rolename* profile.

Add or change any of the following security attributes in the Custom *rolename* profile:

- Commands or actions with security attributes
- Authorizations
- Profile

5. Double-click the Administrative Roles tool, select the role, and choose **Action->Assign Rights**.

If you modified an existing Custom *rolename* Role profile, make sure that it is assigned to the role. The rights profile should be ordered before the `All` profile.

For example, if you are changing the System Administrator role, and the Custom Admin Role profile is not assigned to the role, you would use the Administrative Roles tool to add the Custom Admin Role to the list of Rights for the role. You would also move the Custom Admin Role before the `All` profile in the list.

▼ To Configure a New Role

1. Define the role's responsibilities, and decide what commands, actions, security attributes, and authorizations the role needs to do its work.
2. Decide whether any of the commands or actions need privileges or other security attributes to do their work, and, if so, decide whether the role and the command or action can use these security attributes in a trustworthy manner.
3. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.
4. Bring up the SMC in the desired scope and click the Users tool. Supply a password when prompted.
5. If the role needs to have a new or modified rights profile, double-click the Rights tool to create or modify the rights profile.
See "To Create a Help File for a Rights Profile" on page 89 and "To Create a Rights Profile" on page 89 if you need to create a new rights profile.
To modify a right, select it and follow the online help.
6. Double-click the Administrative Roles tool, and choose Action->Add Role.
Refer to the online help when naming and describing the role.
7. Order the Custom *rolename* Role profile before other profiles you assign to the role.
For example, you would order a Custom Auditadmin Role before the All profile.
8. If you are running the NIS+ naming service, make an entry for the new role in the NIS+ admin group.
See "To Enable a Role to Administer NIS+" on page 102, if needed.

▼ To Enable a Role to Administer NIS+

1. Log into the NIS+ master and assume the Security Administrator role.
2. Double-click the Add to NIS+ Administrative Group action in the System_Admin folder in the Application Manager.
3. To enable a new role to administer NIS+, add the role to the NIS+ admin group.
Use your domain name with the format *subdomain.domain.suffix..*. For example:
Group Name: `admin`
Principal Name: `rolename.security.example.com.`

Note – Remember to type a period (.) at the end of the the domain name.

4. Close the Add to NIS+ Group dialog box.

For example:

```
Group "rolename.security.example.com." created.  
*** Select Close or Exit from the window menu to close this window ***
```

▼ To Enable Remote Role Assumption from Untrusted Systems

Untrusted systems are computers running an operating environment other than the Trusted Solaris operating environment.

Do this procedure on the Trusted Solaris machine where you want to be able to log in remotely.

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.

2. Stop the `init.wbem` program if it is running.

```
$ /etc/init.d/init.wbem stop
```

3. Use the Admin Editor action to open the `/usr/sadm/lib/smc/bin/smcwbemserver` file for editing.

4. Modify the following line to include the `-u` option

```
com.sun.management.viperimpl.server.ViperWbemServer "$@" |&  
>  
com.sun.management.viperimpl.server.ViperWbemServer -u "$@" |&
```

When the `-u` option is specified, after being authenticated the user is presented with a list of the roles that user can assume that are available on the server. The user can then choose a role, enter the role's password, and select the Login as Role button.

5. Save and quit the file.

6. Restart `init.wbem`.

```
$ /etc/init.d/init.wbem start
```


Managing Mail

This chapter covers security, multilevel mailers, and trouble-shooting differences when administering mail in the Trusted Solaris environment.

Note – The Solaris 8 4/01 release contains a number of changes to the `sendmail` program. See “Mail Services” in *Solaris 8 System Administration Supplement* for more information. The Solaris 8 4/01 release is the basis of this Trusted Solaris release.

This chapter contains the following procedures:

- “Managing Mail (Tasks)” on page 110
- “To Configure Users To Receive Mail Below Their Minimum Labels” on page 110
- “To Modify a Mail Alias” on page 111
- “To Permit Users to See the Mail Queue” on page 111
- “To Troubleshoot Mail Delivery Difficulties” on page 112
- “To Trace sendmail for Trusted Solaris Information” on page 112
- “To Check Network Connections for Sending Mail” on page 113
- “To Troubleshoot Loss of Mail Icons” on page 116
- “To Create a Multilevel Action for the Alternate Mail Application ” on page 116
- “To Substitute an Alternate Mail Application for All Users ” on page 118
- “To Install an Alternate Mailer in the Front Panel” on page 119

Managing Trusted Solaris Mail Features

In the Trusted Solaris environment, the System Administrator role sets up and administers mail servers according to instructions in the *Solaris System Administration Guide, Volume 2* and *System Administration Guide, Volume 3*. In addition, the security

administrator determines how Trusted Solaris mail features should be configured. The following sections describe aspects of managing mail that are specific to the Trusted Solaris environment.

.mailrc Is at User's Minimum Label Only

By default, users' `.mailrc` files are stored only in the SLD at the user's minimum label. Users who work at multiple labels do not have a `.mailrc` at the higher labels unless they copy or link the `.mailrc` file to each higher SLD.

The Security Administrator role or the individual user can add the `.mailrc` file to either `.copy_files` or `.link_files`. See `updatehome(1M)` for a description of `.copy_files` and `.link_files`. See "Managing Initialization Files" on page 67 for more information.

For background about mail aliases, see the Mail Aliases section in "Introduction to Mail Services" in the *Solaris System Administration Guide, Volume 3*.

The Solaris Management Console Manages Mail Aliases

Local and name service mail aliases are managed using the Solaris Management Console (SMC) Mailing Lists tool. Depending on the scope of the selected SMC toolbox, an administrator can update the local `/etc/aliases` file, the `mail.aliases` NIS map, or the `mail_aliases` NIS+ table.

Users Cannot Read Email Below Minimum Label

The `sendmail.cf` file has been extended with Trusted Solaris options to enable the security administrator to customize labeled mail delivery. By default, `ADMIN_LOW`-labeled mail is upgraded to the recipient's minimum label. Other mail that is labeled below the recipient's minimum label is returned. `ADMIN_LOW` mail is treated differently from other mail because `ADMIN_LOW` mail is always sent by a system process to an account (usually an administrative role account) that should see the mail.

The default behavior is shown in the commented-out lines in the `sendmail.cf` file.

```
#O LabelAdminLow=upgrade
#O LabelTooLow=return
```

The Security Administrator role may change the values for the Trusted Solaris-specific options in the `sendmail(1M)` configuration file `sendmail.cf` to be consistent with the site's security policy. A user who is cleared to a particular label, such as `CONFIDENTIAL` or `INTERNAL USE ONLY`, should probably not be able to send mail to a user whose minimum label dominates the first user's label, such as `SECRET` or `NEED TO KNOW`.

Users Cannot List the Mail Queue

By default, a user is not able to list queued mail sent by other users. The `restrictmailq` privacy option is set by default in the `sendmail.cf` file.

Listing of the mail queue is done either by entering the `mailq` command or the equivalent command, `sendmail` with the `-bp` option. These commands are in the Mail Management profile, and show mail only at labels dominated by the calling process.

See "To Permit Users to See the Mail Queue" on page 111 for how to enable a user on a particular system to list the queue.

dtmail is the Default Mail Application

By default, `dtmail` is the mail application that is launched from the Mailer subpanel on the Trusted Solaris Front Panel. Trusted Solaris software enables the System Administrator role to substitute an alternate mail application that provides full multilevel mail capabilities.

Without administrative intervention, any user can drag and drop an action for an alternate mail application into the Front Panel and then access the newly-installed mailer at the label of the current workspace. However, since mail monitoring at multiple labels does not occur when an action is installed this way, dragging and dropping by individual accounts of alternate mail actions into the Front Panel is only appropriate at a site using a single label.

Before an alternate mail action can be installed in the front panel, an application must first be defined for the mail application. The example in "To Create a Multilevel Action for the Alternate Mail Application" on page 116 shows the substitution of the OpenWindows mailtool for `Dtmail`, even though it is unlikely that this substitution would be made. The example relies on a predefined OpenWindows mailtool action in the `/usr/dt/appconfig/types/C/sunOW.dt` file as shown below.

```
ACTION OWmailtool
{
    LABEL          OW Mail Tool
    ICON           OWmailtool
```

```

        TYPE          COMMAND
WINDOW_TYPE        NO_STDIO
EXEC_STRING         /usr/openwin/bin/mailtool
    }

```

See “To Create a Multilevel Action for the Alternate Mail Application ” on page 116 for creating an alternate Front Panel mail application, and “To Substitute an Alternate Mail Application for All Users ” on page 118 and “To Install an Alternate Mailer in the Front Panel” on page 119 for different distribution methods.

Troubleshooting Mail Problems

Tracing Mail Delivery Difficulties

Trusted Solaris 8 4/01 software checks host and user labels before sending or forwarding mail.

- The software checks that the mail is within the accreditation range of that host, as described in the following list and in Chapter 8.
- The software checks that the mail is between the account’s clearance and minimum label.

See “To Troubleshoot Mail Delivery Difficulties” on page 112 for specific guidelines in debugging label difficulties.

Tracing sendmail’s Activities

Multiple instances of `sendmail` are involved in local and remote mail delivery. To aid in debugging any problems with `sendmail`, Figure 6–1 shows how data flows through the `sendmail` processes.

Any mailer that is used to send mail (the default is `dtmail`) starts an instance of `sendmail`. This instance of `sendmail` attempts to deliver any mail that originates on the host, storing it in the local `/var/spool/mqueue` MLD until it is delivered [1 in Figure 6–1 shows this instance of `sendmail`]. Normally the message is delivered right away so its stay in the queue is only a matter of seconds. However, if the remote host is down, mail can stay in the queue indefinitely.

An instance of the `sendmail` program also starts when the system is booted. This instance of `sendmail` listens at port 25 and attempts to deliver any mail that it

receives from a remote host, also storing each message in the mail queue MLD until it is delivered [3 and 5 in the example].

Yet another instance of `sendmail` periodically scans the mail queue and attempts to deliver any mail in the queue [2 and 4 in the example]. The following figure shows some of the `sendmail` processes on three hosts: cascade, trustworthy, and juggle. Host trustworthy is the mail relay host for juggle.

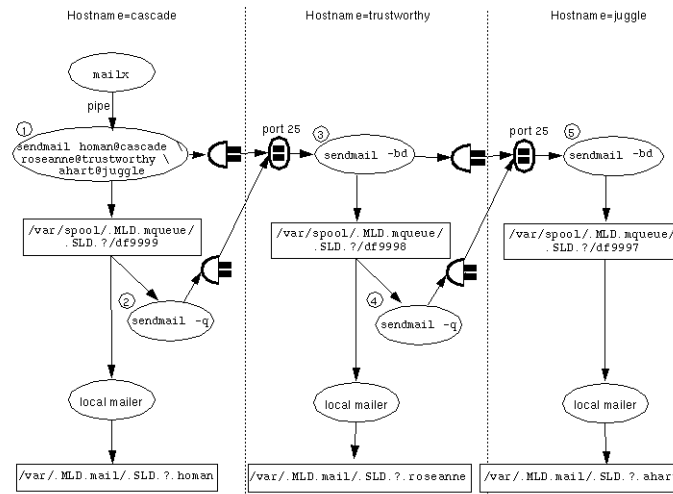


FIGURE 6-1 Sendmail Data Flow Example

When mail is sent to `username@hostname` and `hostname` is a remote host, `sendmail` forwards the message to port 25 of `hostname`. In the example, when mail addressed to `homan@cascade` is sent from another account on host cascade, `sendmail` #1 puts the mail into an SLD within the `/var/spool/.MLD.mqueue` on cascade, where it is delivered by a local mailer. `sendmail` #2 on cascade periodically polls the queue and delivers mail that could not get delivered right away. `sendmail` #3 and #5 on hosts trustworthy and juggle listen on port 25 for incoming mail. The messages originating on cascade that are addressed to hosts trustworthy and juggle are both put into the local `/var/spool/.MLD.mqueue` and sent to port 25 of trusted, which is acting as a mail relay host in this example. The `sendmail` #3 on trusted also puts both messages into an SLD within the local `/var/spool/mqueue`, where the message to `roseanne@trustworthy` is delivered by the local mailer and the message to `ahart@juggle` is forwarded to `sendmail` #5, which is listening at port 25 of juggle.

Debugging sendmail

Debugging sendmail using the `-d` option is described in the *sendmail Nutshell Handbook* published by O'Reilly & Associates, Inc. To review briefly, you can get debugging information by specifying sendmail with the `-d` option followed by *X*. To limit the output of `sendmail -d` to a specific aspect of sendmail's behavior, you can specify a *category* optionally followed by a *dot* (.) followed by a *level* from 0-9, with 9 meaning the most information. Category 75, which is unique to the Trusted Solaris version of sendmail, selects Trusted Solaris debugging information.

Managing Mail (Tasks)

▼ To Enable the IMAP Server to Authenticate Users

This procedure enables the mail server in a NIS+ domain to authenticate users, while protecting the NIS+ password table from being read.

1. On the NIS+ master, assume the System Administrator role and go to an **ADMIN_LOW** workspace.
2. Open the **System_Admin** folder in the Application Manager.
3. Double-click the **Add to NIS+ Administrative Group** action and enter the IMAP server in its full principal name.

The principal name has the format *hostname.subdomain.domain.suffix..* For example:

Group Name: **admin**

Principal Name: **pigeon.aviary.eco.org.**

Note – Remember to type a period (.) at the end of the domain name.

▼ To Configure Users To Receive Mail Below Their Minimum Labels

The value **upgrade** means to upgrade a low-labeled message to the recipient's minimum label, and deliver it.

The value **accept** means to leave the message at the low label and deliver it.

The value **return** means to return the message to the sender (the default).

1. Assume the Security Administrator role and go to an **ADMIN_LOW** workspace.
2. Use the Set Mail Options action in the System_Admin folder in the Application Manager to open the `sendmail.cf` file for editing.
3. Search for TSOL, and change the default settings to reflect site security policy.

```
# TSOL: Incoming mail below recipient's minimum label
# Possible values are return, upgrade, or accept
#O LabelTooLow=return
# Special case for mail labeled admin_low
#O LabelAdminLow=upgrade
```

▼ To Modify a Mail Alias

1. Assume the System Administrator role and go to an **ADMIN_LOW** workspace.
2. Launch the SMC, open the Trusted Solaris Configuration toolbox, and choose the appropriate name service scope for your site (NIS or NIS+).
3. Double-click the Users tool and enter the role password when prompted.
4. Double-click the Mailing Lists tool, and follow the online help for creating and modifying mailing lists.

▼ To Permit Users to See the Mail Queue

1. Assume the Security Administrator role and go to an **ADMIN_LOW** workspace on the system where you want a user to be able to list the mail queue.
2. Use the Set Mail Options action in the System_Admin folder in the Application Manager to open the `sendmail.cf` file for editing.
3. Search for the `restrictmailq` option in the file.

```
# Privacy flags
O PrivacyOptions=authwarnings,restrictmailq
```

4. Remove the `restrictmailq` option.

```
# Privacy flags
O PrivacyOptions=authwarnings
```

5. Save and quit the file.

6. **Open the SMC at the appropriate scope and assign the Mail Management rights profile to the user.**

The user can now use the `mailq` and `sendmail -bp` commands show the mail queue at labels dominated by the user's process.

▼ To Troubleshoot Mail Delivery Difficulties

1. **Check that there is a properly configured network connection between the sending and receiving hosts, as detailed in "To Check Network Connections for Sending Mail" on page 113.**

2. **Check the `nsswitch.conf` file and the mail aliases repositories.**

`sendmail` consults the local `/aliases` file, the NIS map `mail.aliases`, or the NIS+ `mail_aliases` table when determining where to deliver mail. Which alias file it consults depends on the `nsswitch.conf(4)` entry for aliases.

For example, mail to `janez` from a process on her Trusted Solaris desktop `tern` would not go to `janez@tern` if `sendmail` consults the NIS+ `mail_aliases` table and finds an alias of `janez@egret` in that table for user `janez`.

▼ To Trace sendmail for Trusted Solaris Information

1. **Assume the System Administrator role and go to an `ADMIN_LOW` workspace.**
2. **Go to the `/etc/init.d` directory and stop `sendmail`.**

```
$ cd /etc/rc2.d
$ sendmail stop
```

3. **Debug `sendmail` using the `sendmail -d` command followed by the category 75 optionally followed by a dot (`.`) and a level, followed by a space and the address, followed by a message.**

A message can be included either by redirecting the contents of a file to the address, as shown below, or by entering return at the end of the line. In the latter case, a Subject: prompt comes up; after entering the subject, you can create a message from the command line, using the syntax of the `mail(1)` command.

```
$ /usr/lib/sendmail -d75.9 janez@tern < /etc/motd
```

4. **Review the error messages.**
5. **Restart `sendmail` when you are through.**

```
$ cd /etc/init.d
$ sendmail start
```


6. Return to step 7 in “To Check Network Connections for Sending Mail” on page 113 if the user still has trouble sending or receiving mail.

▼ To Check Network Connections for Sending Mail

1. As a user, send mail using the `mailx` command.

```
tern% mailx -v somebody@somehost
Subject: test1
test1
.
```

Review the messages from `mailx`.

2. Log in to the sending host or, if the mail server is not the same as the sending host, log in to the mail server at the label at which the user sends mail.
3. Use the `telnet` command to connect to port 25 of the receiving host.

```
egret% telnet hostname 25
```

If the connection is properly set up, that is, the trusted networking databases for the sending and the receiving hosts have the correct labels, the `sendmail` on the destination host prints a message like:

```
220 hostname Sendmail version ready at date
End the connection by typing quit.
```

```
quit
```

- If the connection seems to be set up properly, go to the following step.
- If `telnet` sends an error message, then the connection is not properly set up. Use the following table to determine the next step.

Type of host	Go to ...
Trusted Solaris host	step 7 and step 8
label-cognizant non-Trusted Solaris host	step 9
unlabeled host (such as Solaris)	step 10

4. Assume a role with the Mail Management right.
5. At the label of the outgoing mail, list the mail queue on the sending host or, if the mail server is not the same as the sending host, list the mail queue on the mail server.

```
$ mailq | more
```

Check the list to see if the mail is stuck on the mail server.

6. **Try the procedure under “To Trace sendmail for Trusted Solaris Information” on page 112.**
7. **If the destination host is running a Trusted Solaris 2.5.1 or later release, do these steps to make sure the destined user is able to receive mail within Trusted Solaris security policy:**
 - a. **Check that the recipient has a valid user account.**

In the Trusted Solaris 8 and Trusted Solaris 8 4/01 releases, use the SMC User Accounts tool. In Trusted Solaris 2.5.1 and Trusted Solaris 7, use the Solstice User Manager.
 - b. **Note the account’s minimum label and clearance.**
 - c. **Check that the label of the mail is within the System Accreditation range of the destination host as specified in the `label_encodings(4)` file.**

`sendmail` does not deliver mail if the label of the mail is outside the System Accreditation Range.
 - d. **Check that the label of the mail is within the User Accreditation Range of the destination host as specified in the `label_encodings(4)` file.**

If the label of the mail is inside the System Accreditation Range but outside the User Accreditation Range, such as mail sent at `ADMIN_LOW` and `ADMIN_HIGH`, go to step 8.
 - e. **Suggested fix:**
 1. If the label of the mail being sent is not in the recipient’s label range, try to find a mutually-acceptable label for the sender and the recipient. If one is found, change the label and try again.
 2. If the mail goes through, instruct the sender to send mail to that recipient at the mutually-acceptable label.
 - f. **If the mail is below the minimum label of the recipient, change the default Trusted Solaris options in the `sendmail.cf` file, if doing so is consistent with your site’s security policy.**

See “Users Cannot Read Email Below Minimum Label” on page 106 and “To Configure Users To Receive Mail Below Their Minimum Labels” on page 110.
 - g. **To enable anyone to receive mail from system processes outside the User Accreditation Range if the `tsoladminlowaccept` or `tsolotherlowreturn` option are used, use the Rights tool to give the user the `solaris.label.range` authorization.**

The default administrative roles have the needed authorization in their profiles.

8. For a destination host running the Trusted Solaris operating environment, check that the sending host has properly configured `tnrhdb` and `tnrhtp` entries for the receiving host.

Note – You can use the `tninfo(1M)` command to check the `tnrhdb(4)/tnrhtp(4)` configuration. The `-h hostname` option lists the name of the template assigned to the specified host, while the `-t template_name` option lists the entries specified in the template, including the host type.

- a. Check that the destination host has the correct template name assigned to it in the `tnrhdb` database, and that the template in the `tnrhtp` file correctly specifies `sun_tsol` as the host type.
 - b. Check that the minimum and maximum label set in the assigned template in `tnrhtp` allow communications at the label of the mail that is not being delivered.
 - c. Once these checks are passed, try step 3 in “To Check Network Connections for Sending Mail” on page 113 to confirm that the network connection works.
9. For a labeled destination host that is not a Trusted Solaris system, check that the sending host has properly configured `tnrhdb/tnrhtp` entries for the receiving host.

Read the `tnrhtp(4)` man page if necessary to find out the correct host type and other options to specify in the template assigned to the host. For example, CIPSO type hosts require certain options, and RIPS0 type hosts require other options.

 - a. Create a template or use an appropriate one in the `tnrhtp`, and check that the correct template is assigned to the host in the `tnrhdb` database.

Double-check the attributes in the template, for example, host type and labe range.
 - b. Once these checks are passed, try step 3 in “To Check Network Connections for Sending Mail” on page 113 to confirm that the network connection works.
10. If the destination host is running an unlabeled operating system, check that the sending host has properly configured `tnrhdb/tnrhtp` entries for the receiving host.
 - a. Check that the destination host has been assigned the correct template name in the `tnrhdb` database, and that the template correctly defines the host’s type as `unlabeled`.
 - b. Check that the default label for the unlabeled host in the assigned template in the `tnrhtp` allows communications at the label of the mail that is not being delivered.

- c. Once these checks are passed, try step 3 in “To Check Network Connections for Sending Mail” on page 113 to confirm that the network connection works.

▼ To Troubleshoot Loss of Mail Icons

If all mail icons disappear from the Front Panel, replace them from the `$HOME/.dt/fp.dynamics` directory.

1. Assume the System Administrator role and go to an `ADMIN_HIGH` workspace.
2. Investigate the account's `.dt/fp.dynamics` directory in its home directory.
During the operation of the system, all changes to the Front Panel are stored in each account's `.dt/fp.dynamics` directory at the session clearance.
3. Copy the contents of `fp.dynamics` to a backup directory and restore the files one by one until the Front Panel configuration is restored.

▼ To Create a Multilevel Action for the Alternate Mail Application

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.
2. Use the Admin Editor action from the `System_Admin` folder to bring up the `/usr/dt/appconfig/types/C/dtwm.fp` file to edit.
3. Find the control section for mail shown below.

```
CONTROL Mail
{
  TYPE                icon
  CONTAINER_NAME      Top
  CONTAINER_TYPE      BOX
  POSITION_HINTS       5
  ICON                DTmail
  LABEL               Mail
  ALTERNATE_ICON       DtMnew
  MONITOR_TYPE         mail
  DROP_ACTION          Compose
  PUSH_ACTION          DTWmail
  PUSH_RECALL          true
  CLIENT_NAME          dtmail
  HELP_TOPIC           FPOnItemMail
  HELP_VOLUME          FPanel
}
```

4. Copy the control text to a file whose name has the .fp extension, for example, mail.fp, and quit the dtwm.fp file.
5. Bring up the Admin Editor action from the System_Admin folder and open the new mail.fp file for editing.
6. Change the title of the mail control to OW_Mail.

```
CONTROL OW_Mail
```

7. Change the following variables to the following values:

```
ICON          OWmailtool
LABEL         OW Mail Tool

PUSH_ACTION   OWmailtool

CLIENT_NAME   mailtool
```

The ICON field identifies the icon of the replacement application.

The LABEL field changes the icon label that appears with the icon of the replacement application.

The PUSH_ACTION field identifies the replacement action to be run when the user clicks on the new mail icon. The action name supplied here must be defined in the one of the application search paths. The OWmailtool action shown is defined in sunOW.dt in the /usr/dt/appconfig/types/locale directory.

The CLIENT_NAME field identifies the executable for the replacement application. The path for CLIENT_NAME must be defined by an EXEC_STRING in the action's definition. For example, the OWmailtool action has EXEC_STRING defined as /usr/openwin/bin/mailtool.

8. Change DROP_ACTION or leave as shown below.

```
DROP_ACTION    Compose
```

Other mailers may or may not have a Compose action. For example, OpenWindows mailtool does not. If you leave the DROP_ACTION as Compose, if someone drags mail to the mail icon, a dtmail Compose window will come up. If you remove the DROP_ACTION, nothing happens if mail is dragged to the mail icon.

9. Leave the rest of the variables unchanged, as shown below.

```
TYPE           icon
CONTAINER_NAME Top
CONTAINER_TYPE BOX
POSITION_HINTS 5

ALTERNATE_ICON DtMnew
MONITOR_TYPE   mail

PUSH_RECALL    true
```

```

HELP_TOPIC          FPOnItemMail
HELP_VOLUME         FPanel

```

When `PUSH_RECALL` is `true`, an application that is launched for a second time uses an existing application window.

10. **Save the changes and quit the file.**

11. **Place the `mail.fp` file, in an accessible directory, such as `$HOME/secadmin`.**

```
$ mv mail.fp /home/secadmin/cde_changes/
```

▼ To Substitute an Alternate Mail Application for All Users



Caution – Do this procedure on every system before users start getting mail. If you do it later, you will need to clean up the contents of directories created by the window system in every `.dt/fp.dynamics` directory in every SLD in every home directory MLD.

1. **Make sure that the alternate mail action has been fully tested.**

2. **Assume the System Administrator role and go to an `ADMIN_LOW` workspace.**

3. **In a terminal, go to the `/etc/init.d` directory and stop `sendmail`.**

```
$ cd /etc/init.d
$ sendmail stop
```

4. **Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.**

5. **Use the Admin Editor to replace the `CONTROL Mail` section of the `/usr/dt/appconfig/types/locale/dtwm.fp` file with the contents of the `mail.fp` file.**

Replace —

```

CONTROL Mail
{
    TYPE          icon
    CONTAINER_NAME Top
    CONTAINER_TYPE BOX
    POSITION_HINTS 5
    ICON          DTmail
    LABEL         Mail
    ALTERNATE_ICON DtMnew
    MONITOR_TYPE  mail
    DROP_ACTION   Compose
    PUSH_ACTION   DTWmail

```

```

PUSH_RECALL      true
CLIENT_NAME      dtmail
HELP_TOPIC       FPOnItemMail
HELP_VOLUME      FPanel
}

```

With the contents of the mail.fp file —

```

CONTROL OW_Mail
{
  TYPE              icon
  CONTAINER_NAME    Top
  CONTAINER_TYPE    BOX
  POSITION_HINTS     5
  ICON              OWmailtool
  LABEL             OW Mail Tool
  ALTERNATE_ICON    DtMnew
  MONITOR_TYPE      mail
  DROP_ACTION       Compose
  PUSH_ACTION       OWmailtool
  PUSH_RECALL       true
  CLIENT_NAME       mailtool
  HELP_TOPIC        FPOnItemMail
  HELP_VOLUME       FPanel
}

```

6. Change the name of the CONTROL back to Mail and save the dtwm.fp file.

```

CONTROL OWMail
:wq

```

7. If the system has been receiving mail, remove all contents of the `$HOME/.dt/fp.dynamics` directory.
8. Restart the Workspace Manager from the workspace menu to see the changes to the dtwm.fp go into effect in the Front Panel.
9. Assume the System Administrator role and go to an ADMIN_LOW workspace.
10. In a terminal emulator such as dtterm, restart sendmail.

```

$ cd /etc/init.d
$ sendmail start

```

▼ To Install an Alternate Mailer in the Front Panel

1. Assume the System Administrator role and go to an ADMIN_LOW workspace on the system where you want to install an alternate mail program.

2. In a terminal, go to the `/etc/init.d` directory and stop `sendmail`.

```
$ cd /etc/init.d
$ sendmail stop
```

3. Using the File Manager, change to the directory where the alternate mail application's control file (`mail.fp`) resides.

The Security Administrator placed it in an accessible directory in step 11 of "To Create a Multilevel Action for the Alternate Mail Application " on page 116.

4. Add `mail.fp` to the `/usr/dt/appconfig/types/locale` or `/etc/dt/appconfig/types/locale` directory.

5. Go to a user workspace.

6. Click the Mailer subpanel access button to bring up the subpanel.

7. Drag the icon for the alternate mailer's front panel control file to the Install Icon dropsite in the Mailer subpanel.

The icon for the alternate mail application should appear in the Mail slider.

8. Click the right mouse button while the pointer is over the alternate mail and select Copy to Main Panel.

9. Remove each old mail icon in the subpanel by clicking the right mouse button over an icon for the old application and selecting Delete.

Note – Remove all old icons. You cannot have a mixture of mail applications running at the same time.

10. Select Restart Workspace Manager from the Workspace Menu to adjust the size of the subpanel.

11. Return to the System Administrator workspace and restart `sendmail`.

```
$ cd /etc/init.d
$ sendmail start
```

12. If this is an end user system, delete the System Administrator workspace.

Managing Computers and Networks

This chapter describes in detail the concepts and goals of trusted networking. For an overview of trusted networking, see “Administering Trusted Networking” in *Trusted Solaris Administration Overview*.

Managing Trusted Network Communications

The Trusted Solaris operating environment supports network communications between Trusted Solaris computers and any of the following types of computers:

- Other computers running the Trusted Solaris operating environment.
- Computers running operating environments that do not recognize security attributes but do support TCP/IP (such as Solaris and other UNIX systems, Windows, and MacOS)
- Computers running other trusted operating systems that recognize some of the Trusted Solaris security attributes

Network communications and services are managed by several Trusted Solaris subsystems.

- Trusted NFS manages mounts of file systems from other computers.
Mounts among Trusted Solaris computers and other computers that recognize NFS are supported. See Chapter 9 for more on mounting.
- NIS and NIS+ provide centralized management of configuration files defining hosts, networks and users.
See Chapter 10 for NIS and NIS+ differences in the Trusted Solaris environment.

- The Solaris Management Console is used to centrally manage users, computers, and networks.

The *Trusted Solaris Installation and Configuration* guide describes how to add new workstations and servers during configuration of a distributed system. See Chapter 8 for additional details about how to set up security attributes for computers.

- Trusted networking and extended routing software supports trusted network communications.

SMC Tools for Administering Computers and Networks

The SMC Trusted Solaris Configuration toolbox contains the following tools for configuring computers and networks and defining security attributes for computers, networks, and network interfaces:

- The Computers and Networks tool, which includes:
 - The Computers tool for adding new computers
 - The Add Network option for specifying netmasks
 - The Trusted Solaris Security Families tool for assigning security attributes to computers
- The Trusted Solaris Interface Manager tool for assigning security attributes to network interfaces (only needed if the default values are not appropriate)

Note – The Interface Manager modifies the local `/etc/security/tnidb` file, and the tool displays only when the scope of the selected toolbox is Files.

The Computers and Networks and the Interface Manager tools are shown in the following figure.

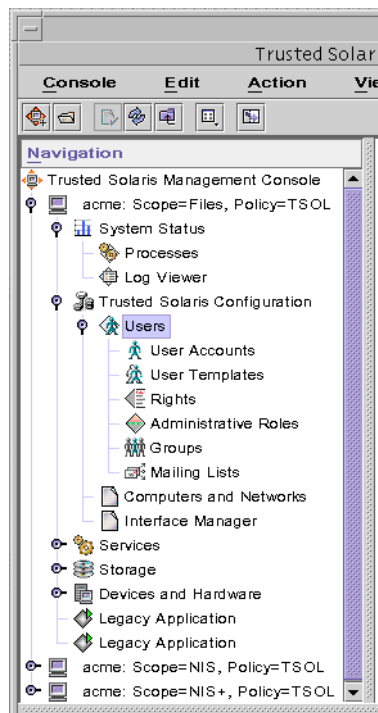


FIGURE 7-1 Solaris Management Console Tools

Meeting the Goals of Trusted Networking

The overall goal of trusted networking software is to ensure that the Trusted Solaris security policy is enforced even when the subjects (processes) and objects (data) are located on different hosts.

Assigning security attributes to computers and networks and to mounted file systems ensures the following:

- Data in network communications is properly labeled.
- Mandatory access control (MAC) rules are enforced when data is sent or received across a local network and when file systems are mounted.
- MAC rules are enforced when data is routed to distant networks.

Understanding Security Attributes Assigned to Computers

Security attributes are administratively assigned to computers (hosts and routers) by means of templates. The Security Administrator role administers templates and assigns templates to computers using the Security Families tool. If a computer does not have a template assigned, no communications are allowed with that computer. Computers that share the same template are considered to be part of the same security family.

Every template has a `Host Type`, which determines which protocol is used to communicate with the computer that is assigned the template. The protocols tell the kernel which security attributes to look for in the header of an incoming packet or to insert into an outgoing packet. See “Host Types” on page 125.

Every template also has an Accreditation Range (consisting of a Minimum Label and a Maximum Label) and a default DOI (Domain of Interpretation). For details about these attributes, see “Computer Accreditation Range” on page 126 and “Domain of Interpretation (DOI)” on page 126.

Each host type has its own set of additional required and optional security attributes, which are introduced in the following list:

- Templates for the Unlabeled and RIPS0 host types specify a Default Label that is used to control communications with computers whose operating environment is not aware of labels, such as Solaris or RIPS0-cognizant operating environments.
Because communications with these computers are essentially limited to the Default Label, they are referred to as single-label computers. See “Default Label” on page 127.
- Templates for single-label computers also specify a Default Clearance. One or more optional privileges can be specified in the template’s Forced Privileges field. See “Default Clearance” on page 127 and “Forced Privileges” on page 127.
- The template for the Trusted Solaris host type has an Allowed Privileges field that can optionally be used to limit the privileges accepted from the remote computer. See “Allowed Privileges” on page 128.
- The template for any host type can be used to specify an IP label to be used in trusted routing of packets. See “Using IP Labels in Trusted Routing” on page 129.

For more about specifying an IP label or how to change the default DOI, see “Advanced Security Attributes” on page 128.

Host Types

The following table describes the host types for which entries can be made in the trusted network databases. The first column shows the name used in the Security Families host type menu.

TABLE 7-1 Host Types, Protocols, and Notes

Name in Template Manager	Protocols and Notes
Trusted Solaris	The TSOL protocol simplifies passing security attributes between computers running Trusted Solaris 2.5.1, Trusted Solaris 7, Trusted Solaris 8, or Trusted Solaris 8 4/01 releases. TSOL is a derivative of the TSIX(RE) 1.1 - SAMP protocol that passes the security attributes in a similar place in the network protocol stack and uses similar header structures. The TSOL protocol passes security attributes in binary form and thus does not require token mapping. NOTE: For communications between Trusted Solaris computers, either the Trusted Solaris or TSIX host type can be assigned in the templates, depending on whether you want the labels to be transmitted in binary form or in token form. If only the labels' names differ on two computers while the labels' binary representations are the same, the Trusted Solaris host type can be used. If the labels' names are the same but the labels' binary representations are different on both Trusted Solaris computers, the TSIX host type can be assigned.
Unlabeled	This host type is assigned to computers running Solaris or other unlabeled operating environments to specify a default label and default clearance to apply to communications with the unlabeled computer. Also, a minimum and maximum label can be set to allow the sending of packets to an unlabeled gateway for forwarding when the packets' labels do not match the default label and would therefore not be sent to the computer as a destination.
RIPSO	Revised IP Security Option (RIPSO) described in the IETF RFC 1108. It specifies a DoD IP labeling method to incorporate labels into IP packets, which are then used for network mandatory access control checks. A fixed RIPSO label specified in the template is applied to network packets interchanged with the particular host. Though this functionality does not fully meet the RFC specifications, it is expected to supply sufficient functionality where RIPSO labels are needed.
CIPSO	Common IP Security Option (CIPSO) protocol TSIX(RE) 1.1 is used to specify security labels that are passed in the IP options field. CIPSO labels are derived automatically from the data's label. Tag type 1 is used to pass the CIPSO security label. This label is then used to make security checks at the IP level and to label the data in the network packet.

TABLE 7-1 Host Types, Protocols, and Notes (Continued)

Name in Template Manager	Protocols and Notes
TSIX	Trusted Security Information Exchange for Restricted Environments (TSIX/RE) protocol uses token mapping to pass security attributes. Can be used for computers running the Trusted Solaris or other TSIX-cognizant operating environments. See the NOTE for the Trusted Solaris host type in the first entry in this table.

Computer Accreditation Range

The Minimum Label and the Maximum Label are used in the following ways:

- To set the range of labels that can be used when communicating with a computer.
In order for a packet to be sent to a computer, the label of the packet must be within the label range assigned to the destination computer in its template.
- To set a label range for packets being forwarded through an unlabeled or RIPS0 gateway.
The label range can be specified in the template for an unlabeled or RIPS0 host type to make it possible to forward a packet to that computer for forwarding, even when the packet's label is not the same as the Default Label.

Domain of Interpretation (DOI)

A default Domain of Interpretation is assigned in the default templates for all host types. Two computers need to have the same DOI in order to communicate. Organizations with the same DOI need to agree among themselves about how labels and other security attributes are to be interpreted. Each host type has a DOI associated with it. By default each existing or new template has the default DOI specified in the DOI field. You do need to change the default DOI unless you have reasons for wanting to do so.

As mentioned under "Host Types" on page 125, either the Trusted Solaris or TSIX host type can be specified in templates assigned to Trusted Solaris computers. If the NOTE in the first entry in Table 7-1 is true for your site, the Trusted Solaris or TSIX host-type computers can share the same DOI .

DOIs in Trusted Solaris IPv4 Packets

In Trusted Solaris IPv4 packets, the DOI is carried in the packet along with the label. In an IPv4 packet, the specified DOI is included both with the IP options (if any are specified) and in the SAMP header.

Headers (Options [IP options including DOI])	SAMP including DOI	Data
--	--------------------	------

DOIs in Trusted Solaris IPv6 Packets

Note – Trusted routing using IP labels is not supported with IPv6.

In Trusted Solaris IPv6 packets, label information is carried in multilevel security (MLS) options portion of the packet's Headers. Because label information is in the Headers portion of the packet, the packet's label can be used for routing.

Headers (Options [SAMP MLS options including DOI]	Data
--	------

To specify a DOI other than the default, use the Advanced Security Attributes tabs.

Default Label

Each unlabeled or RIPS0-type computer is assigned a single label in the Default Label field. The Default Label assigned to an unlabeled or RIPS0-type computer should reflect the level of trust that is appropriate for the computer and its users. For RIPS0 hosts, the Default Label should be the same as the RIPS0 Label (which is a combination the RIPS0 Send Class and the RIPS0 Send PAF).

Default Clearance

Each single-label computer (with the Unlabeled or RIPS0 host type) is assigned a clearance in the Default Clearance field. The clearance sets the upper limit for write operations performed on the Trusted Solaris computer by someone on the unlabeled host. For example, on an unlabeled computer with a Default Label of CONFIDENTIAL and Default Clearance of SECRET, a user who is working on a file system mounted from a Trusted Solaris host can open an upgraded file with a label of SECRET and write into it (if the file's name is known to that user).

Forced Privileges

One or more privileges can be specified in the Forced Privileges field of a template that has the unlabeled host type. An unlabeled computer does not recognize

privileges. Specifying privileges in this field affects only how the Trusted Solaris computer handles requests from a program that is running on the unlabeled computer. Privileges can be specified to allow a client from an unlabeled computer to do something not otherwise permitted, such as reading a file whose label dominates that of the client or communicating with X clients owned by another user.

Allowed Privileges

Remote Trusted Solaris computers can usually be trusted to provide correct privileges. If needed, the privileges that a remote Trusted Solaris computer is allowed to use can be controlled by specifying a restricted set of privileges in the Allowed Privileges field of a template with the Trusted Solaris host type. Processes running on a remote Trusted Solaris system communicate their effective privileges as part of their security attributes. You can locally restrict those privileges to the ones that are specified in the Allowed Privileges set.

Advanced Security Attributes

The Advanced Security Attributes tab in the Security Families Template dialog box sets the following options.

- DOI

Every type of supported protocol has a domain of interpretation field. The DOI identifies the labeling scheme. Computers need to have the same DOI in order to communicate. Two organizations that use the same DOI need to agree among themselves to interpret label information the same way.

You need to replace the default domain of interpretation (DOI) only if your site needs another number than the default that is assigned to each host type. Replace the DOI, if desired, by entering an integer into the DOI field.

The type of DOI (TSOL, TSIX, or CIPSO) is determined from the type of host and from any IP label specified in a machine's template. For example, on a Trusted Solaris router with an IP label of CIPSO, the DOI is understood to be a CIPSO DOI.

- IP Options

If using trusted routing with IPv4 packets, choose either "none," "CIPSO," or "RIPSO" from the IP Label pull-down menu.

When the CIPSO IP label is specified in a host's template, then a CIPSO label is inserted into the IP options portion of any packet outgoing to that host. See "CIPSO Labels in Packets" on page 132 for how CIPSO labels are used.

If you choose RIPSO, you need to choose a RIPSO Send Class, an optional RIPSO Send PAF, and RIPSO Return PAF from the pull-down menus. PAF means Protection Authority Flag. Any Send PAF specified is used like a compartment

name along with the classification to make up the RIPS0 label (as in Top_Secret SCI). The PAF specified in the Return PAF is used in labeling ICMP messages that can be generated as errors in response to incoming RIPS0 labeled packets. The Send Class is also sent back with the RIPS0 error in an ICMP message. The RIPS0 label should have the same name as the Default Label assigned to the host. Make sure to specify the same RIPS0 label and RIPS0 PAFs for the sending host, all gateways, and the destination host. See “RIPS0 Labels in Packets” on page 133 for how RIPS0 labels are used.

Using IP Labels in Trusted Routing

If a computer has an IP Label type of RIPS0 or CIPSO specified in its template, the specified type of IP label is put into outgoing packets, and the incoming packets from the specified host must contain an IP label of the specified type. IP labels can be used for trusted routing. Packets with an IP label are only forwarded to routers whose label range allows the specified IP label.

Some organizations have the requirement to label all of their packets with RIPS0 or CIPSO labels, unless the packets are being sent to unlabeled computers directly connected to the network. Others need to use IP labels for trusted routing of packets going to certain destination hosts. In a homogeneous Trusted Solaris security domain, this is accomplished by assigning a template with the Trusted Solaris host type and an IP label of either RIPS0 or CIPSO to all or some Trusted Solaris computers.

Similarly a template with the TSIX host type can also be configured with CIPSO or RIPS0 labels to achieve the same labeling of packets for TSIX hosts.

And, of course, packets to and from a host assigned a template with a CIPSO or RIPS0 host type carry either a CIPSO or RIPS0 IP label. The IP Options supported in the templates for the Unlabeled host type provide a way to label packets coming into a Trusted Solaris security domain from unlabeled computers. Unlabeled packets become labeled when they pass through Trusted Solaris/ripso or Trusted Solaris/cipso gateways on their way to other Trusted Solaris/ripso or Trusted Solaris/cipso computers. The RIPS0 or CIPSO labels are stripped from packets before they are delivered to unlabeled computers, which are typically outside the security domain. To accomplish this, administrators can specify an IP label of RIPS0 or CIPSO in the template for an unlabeled host.

Default Templates

Trusted Solaris software ships with a set of templates that matches the `label_encodings(4)` file on the installation disk. Icons for all defined templates appear when the Security Families tool is double-clicked. The Security Families tool enforces the required fields in the templates, based on the host type.



Caution – If your site has installed its own label encodings, you *must* modify the templates to work with your labels.

- All default templates should be assessed for their applicability and can be used as is or copied, renamed, or modified by the Security Administrator role.
- New templates can be added.

The simplest and safest configuration is to enable communication only among Trusted Solaris computers that share the same `label_encodings` file. To set up such a configuration, the System Administrator role can assign the default TSOL template or other similar template with the Trusted Solaris host type to all Trusted Solaris computers.

Default Templates for Trusted Solaris Systems

A computer running Trusted Solaris 2.5.1 and later compatible releases can be assigned any template that has the Trusted Solaris host type. See the online help in the Security Families tool for a description of the default templates for the Trusted Solaris host type.

Default Templates for Unlabeled or RIPS0 Computers

The Trusted Solaris environment supports communications with computers running operating environments that do not recognize labels (such as the Solaris operating environment). A computer that does not recognize labels or that uses RIPS0 labels must be assigned a single label and a clearance. The label and clearance restrict communications with that computer. Before assigning a template that has the Unlabeled or RIPS0 host type to an unlabeled host, specify the following:

- An appropriate label in the Default Label field.
- An appropriate clearance in the Default Clearance field.

- The Maximum Label equal to the Default Label, unless the unlabeled host is a gateway that needs to forward packets at labels that are not equal to its default label.



Caution – When creating or modifying a template for an unlabeled or RIPS0-type computer, do not forget to change the default label to reflect the level of trust it should be accorded. Administrators who report problems with not being able to communicate with remote single-label computers at the expected label have usually forgotten to specify that label in the Default Label field.

The default unlabeled and ripso host type templates are valid only when either the default label_encodings file is used or another label encodings file with the same label names and binary representations for the labels. See the online help in the Security Families tool for descriptions of the default unlabeled or RIPS0 templates.

Do not use the admin_low template during normal system operations. The admin_low template is needed during initial boot only, before the system is configured. The template assignment is stored in the tnrhtp and tnrhdb databases in /etc/security/tsol on the installed machine. Once the system is configured, the Security Administrator role should either remove the 0.0.0.0 entry entirely or change it to assign a template with an appropriate hots type and security attributes.

Wildcard Entry and Prefix Length

A wildcard IP address is the IP address of a subnetwork. A subnetwork is defined by its IP address and its netmask. The netmask determines the prefix that has to be common to all the addresses belonging to a subnetwork.

For example, the IP address 192.168.123.0 is a wildcard with a netmask = 255.255.255.0. The subnet is made up of all the IP addresses between 192.168.123.1 and 192.168.123.255. A optional Prefix Length can be specified in the form of an integer. The prefix length determines the size of the subnet and is the number of 1 bits in the netmask.

TABLE 7–2 Wildcard Address, Netmask, and Prefix Length

class A addresses: a.0.0.0, or a	class B addresses: a.b.0.0, or a.b	class C addresses: a.b.c.0, or a.b.c
netmask = 255.0.0.0	netmask = 255.255.0.0	netmask = 255.255.255.0
prefix length = 8	prefix length = 16	prefix length = 24

With variable-length subnetting, the prefix length does not have to be a multiple of 8. For example, you can have the IP address 192.168.123.224, with a netmask =

255.255.255.224, and a prefix length = 27, covering the addresses between 192.168.123.225 and 192.168.123.255. IPv4 network addresses can have a prefix length between 1 and 32. IPv6 network addresses can have a prefix length between 1 and 128.

The trusted network software looks first for an entry that specifically assigns the host to a template, and if it does not find a specific entry, the software looks for the subnetwork entry that best matches the hosts's IP address (a subnetwork with the longest prefix length to which that address belongs).

If a computer's IP address cannot be matched to an entry, communication with that computer is not permitted.

A default 0 . 0 . 0 . 0 entry matches all computers that are not otherwise matched by other entries.

Sites that need to strictly control remote access should remove the 0.0.0.0 entry. They should also assess whether to use any wildcard addresses. For more information, see the `tnrhdb(4)` man page.)

CIPSO Labels in Packets

The CIPSO label is derived from the actual label of the data on the sending Trusted Solaris computer.

The trusted networking software puts a CIPSO label and a DOI (domain of interpretation) number into the IP option for outgoing packets and also looks for a CIPSO label and DOI in the IP option of incoming packets, if the trusted network template entry assigned to the remote host meets one of these criteria:

- Assigns the host the CIPSO host type
- Assigns the host the Trusted Solaris host type, setting the IP label type to CIPSO
- Assigns the host the TSIX host type, setting the IP label type to CIPSO

The CIPSO label that is inserted into outgoing packets is derived by the trusted networking software from the actual label associated with the data. Sometimes Trusted Solaris labels match directly to a CIPSO label. For example, the label of CONFIDENTIAL matches the CIPSO label of CONFIDENTIAL. However, most Trusted Solaris labels do not map directly to CIPSO labels.

Note – At a site that plans to use CIPSO labels for trusted routing or wishes to communicate with a host with a host type of CIPSO, the Security Administrator role should plan ahead to configure the site's labels so they map well to CIPSO labels.

A DOI (domain of interpretation) must also be specified, and the same DOI must be:

- Assigned to the sending host
- In a routing table entry for all gateways through which messages travel and understood by routers
- Assigned to the destination host

Ensuring Labels Are Mappable to CIPSO Labels

The Security Administrator role needs to plan ahead to ensure that the labels defined in the `label_encodings(4)` file map well to CIPSO labels. See *Trusted Solaris Label Administration*.

RIPSO Labels in Packets

The RIPSO, Revised IP Security Option, protocol is described in the IETF RFC 1108. The trusted networking software puts a RIPSO label into the IP option for outgoing packets and also looks for a RIPSO label in the IP option of incoming packets from a host, if the trusted network template entry for the host meets one of these criteria:

- Assigns the host the `ripso` host Type
- Assigns the host the `sun_tsol` host type, specifying the IP Label Type as RIPSO
- Assigns the host the `tsix` host Type, specifying the IP Label Type as RIPSO

RIPSO labels on outgoing packets are administratively defined. The Security Administrator role specifies them in the `tnrhtp` database, putting the classification in the RIPSO Send Class field and the compartment(s), or protection authority flags (PAF) in the RIPSO Send PAF field.

The following RIPSO Send classifications are supported: `Top_Secret`, `Secret`, `Confidential`, and `Unclassified`.

The RIPSO Send PAF and Return PAF fields refer to Protection Authority Flags, which are shown in the following table. PAFs specified in the Send PAF field are used like compartment names along with the classification within the RIPSO labels (as in `Top_Secret SCI`). PAFs specified in the Return PAF field are used in labeling ICMP messages that can be generated as errors in response to incoming RIPSO labeled packets. The classification sent back in an ICMP message is the same as the RIPSO classification in the packet.

TABLE 7-3 Protection Authority Flags

Protection Authority Flags (may be specified along with supported classifications in RIPSO labels or specified as RIPSO errors)
GENSER

TABLE 7-3 Protection Authority Flags (Continued)

Protection Authority Flags (may be specified along with supported classifications in RIPS0 labels or specified as RIPS0 errors)

SIOP-ESI

SCI

NSA

DOE

Understanding Security Attributes Assigned to Network Interfaces

All interfaces on a computer running Trusted Solaris software are automatically detected by the trusted network software and assigned a default set of attributes. The Interface Manager shown below is used only when the Security Administrator role wants to change the defaults for an interface.

The default attributes are shown in the following table:

Default Label	Minimum Label	Maximum Label	Default Clearance	Forced Privileges
ADMIN_LOW	ADMIN_LOW	ADMIN_HIGH	ADMIN_HIGH	None

Summary – Any values specified for a computer in a template take precedence over any values supplied for the network interface, and if no values are specified, system defaults apply. For example, if computer A is assigned a default label of INTERNAL, while the network interface that is connected to the network where computer A resides is assigned a default label of PUBLIC, the data coming from computer A is assigned the INTERNAL label. The default label assigned by the network interface is not used.

Network Interface Accreditation Range

The Minimum Label and the Maximum Label are used to set the range of labels for data that can be sent through the interface.

Note – Full communications within a Trusted Solaris domain require an accreditation range of ADMIN_LOW to ADMIN_HIGH.

To be able to leave certain fields empty in a single template assigned to one computer or to a group of computers that is accessed through the same network interface, the Security Administrator role can specify the values in an entry that applies to that network interface.

The entries assigned to network interfaces are looked at only if certain fields are left empty in the template assigned to a computer. If a value is not found either in the template that covers the host or in an entry that applies to the interface through which the remote computer is accessed, then a set of default values is applied.

Note – Restrict the accreditation range on a network interface with care. Network services fail unless the network interface is configured with an accreditation range that includes the labels upon which those services depend. For example, audit clients cannot write ADMIN_HIGH audit data onto the audit server unless the ADMIN_HIGH label is in the range. Full communications within a Trusted Solaris domain require an accreditation range of ADMIN_LOW to ADMIN_HIGH.

Default Security Attributes

The Default Label, Default Clearance, and optional Forced Privileges in the Interface Manager are rarely useful. They would be used when a Trusted Solaris computer is communicating with a computer that is running an operating system that does not recognize labels or privileges, such as the Solaris operating environment, and then only if the same fields have been left empty in the template that applies to the single-label computer. For example, the Security Administrator role might create an entry for a second interface on the local computer that would apply the same label, clearance, and optional forced privileges to all computers running Solaris on the network that is connected to the second interface. These fields could then be left empty in any templates that cover the computers (as specified in the Security Families tool in Computers and Networks).

Default Label	The Default Label should reflect the level of trust that is appropriate for the computer and its users.
Default Clearance	The Default Clearance sets the upper limit for write operations performed on the Trusted Solaris computer by someone on the unlabeled computer. For example, on an unlabeled computer with a Default Label of CONFIDENTIAL and Default Clearance of SECRET, a user who is working on a file system mounted

from a Trusted Solaris computer can open an upgraded file with a label of SECRET and write into it (if the file's name is known to that user).

Forced Privileges An unlabeled computer does not recognize privileges. Specifying privileges in the Forced Privileges field affects only how the Trusted Solaris system handles requests from a program that is running on the unlabeled computer. Specifying privileges enables a client from an unlabeled computer to do something not otherwise permitted, such as reading a file whose label dominates that of the client or communicating with X clients owned by another user. If the corresponding values are set in a template that covers the computer, the value in the template takes precedence over the values specified for the network interface.

The following describes whose values are used for a network interface:

1. Is the needed value specified in a remote host template?
 - a. If yes, the value in the template is used
 - b. If no, is the needed value specified in an entry for the interface?
 - i. If yes, use the value specified for the interface.
 - ii. If no, use the default value.

Accreditation Checks

The trusted networking software performs accreditation checks to compare the security attributes of the source host, the destination host, and of the routes along the way.

Security attributes for the accreditation range check (accreditation range and any CIPSO or RIPS0 label information that may be specified) are obtained from a host's templates. The security attributes for a route (its SRI) are obtained from the route's emetric in the routing table. If an emetric for a route has not been specified, the security attributes of the first hop gateway host's entries are checked.

On a router, accreditation checks are performed only if the packet to be forwarded has RIPS0 or CIPSO labels and then the labels in the IP options portion of the packet are used. If the packet has a CIPSO label, its label is compared to the label range of the incoming and outgoing interface. Its label is also compared to the label range of the next hop gateway.

MAC Enforcement on Outgoing Messages

The following accreditation checks are performed on the sending host.

- The label of the packet being sent must be:
 - Within the accreditation range of the destination host
 - Within the accreditation range of the network interface of the source host.
- If the packet has a CIPSO label, then its DOI must match the DOI of the destination and of the route's emetric. If no emetric is specified for the route, the DOI must match the DOI of the first hop gateway.
- If the packet has a RIPS0 label, then its RIPS0 label and PAF flag must match the RIPS0 label and PAF flag of the destination and of the route's emetric. If no emetric is specified for the route, the RIPS0 label and PAF flag must match the RIPS0 label and PAF flag of the first hop gateway.
- If the destination is specified as a MSIX host, then the label of the packet being sent must be within the accreditation range of the destination host and the route's emetric must include the MSIX attribute. If no emetric is specified for the route, the host type of the first hop gateway must be specified as MSIX and the label of the packet must be within the accreditation range specified for the first hop gateway.

Note – A first hop check occurs when a message is being sent from a host on one network to a host on another through a gateway.

MAC Checks on Messages Being Forwarded

On a Trusted Solaris gateway, accreditation checks are performed for the next hop and for the network interfaces.

If the packet has CIPSO label information, the following must be true for a packet to be forwarded:

- The route's emetric must include the CIPSO option. If no emetric is specified for the route, the next hop gateway's entry must be defined as one of the following:
 - CIPSO host type
 - sun_tsol host type with a CIPSO IP label
 - tsix host type with a CIPSO IP label
- The CIPSO label of the packet must be within the accreditation range from the emetric of the route. If no emetric is specified for the route, the packet's CIPSO label must be within the accreditation range specified in next hop gateway's entry.
- The CIPSO DOI specified in the network database entry for the outgoing interface must equal the packet's DOI.

If the packet has RIPS0 label information, the following must be true for a packet to be forwarded:

- The route's emetric must include the RIPS0 option. If no emetric is specified for the route, the next hop gateway's entry must be defined as either of the following:
 - RIPS0 host type
 - tsol host type with a RIPS0 IP label
 - tsix host type with a RIPS0 IP label
- The RIPS0 label of the packet and PAF must be the same as the RIPS0 label and RIPS0 PAF in the emetric of the route. Or, if no emetric is specified for the route, the packet's RIPS0 label and RIPS0 PAF must be the same as the RIPS0 label and RIPS0 PAF specified in next hop gateway's entry.

If the label of a message is not within the minimum and maximum labels specified in the accreditation range for any of the destination host, gateways, or the network interface, the message is dropped.

MAC Enforcement on Incoming Messages

The following checks are performed on a receiving host.

- The label of the packet being received must be:
 - Within the accreditation range specified in the source host's trusted network database entry
 - Within the accreditation range specified in the trusted network database entry for the network interface receiving the data
- If the packet has a CIPS0 label, then its DOI must match the DOI specified in the receiving host's trusted network database entry.
- If the packet has a RIPS0 label, then its RIPS0 label and PAF flag must match the RIPS0 label and PAF flag specified in the trusted network database entry for the receiving host.

For incoming communications, the Trusted Solaris networking software obtains labels and other security attributes from the packets themselves whenever possible—which is only completely possible when the messages are sent from systems that support labels and all the other required attributes in a form recognized by the Trusted Solaris software. In many cases, packets arrive from hosts that are not label-cognizant or that do not send recognizable labels, or the packets do not have all of the other required attributes in their packets.

When the needed security attributes are not all available from a packet, those that are lacking are assigned to the message from trusted networking databases. Any attributes

not obtainable from the host's entry are supplemented by the attributes specified in the entry in the trusted network interface database entry the interface through which the message arrives.

Administering Routing

Some sites may restrict communications outside of the local network to a single label for publicly-available information, such as UNCLASSIFIED or PUBLIC. These sites specify the desired single label as both the maximum and minimum label assigned to the network interface that is connected to the external network. The Trusted Solaris environment supports additional methods for routing communications between networks, so that the Security Administrator role can set up routes that enforce the degree of security required by the site's security policy. See the *TCP/IP and Data Communications Administration Guide* for more details about TCP/IP and routing.

Background on Routing

For communications sent to destinations on the same subnet, accreditation checks are performed by Trusted Solaris endpoints only since no routers are involved. (Because gateways and routers route packets, the terms gateway and router are used interchangeably in this discussion.) Accreditation range checks are performed at the source. If the receiving host is running Trusted Solaris, accreditation range checks are also performed at the destination.

When the source and destination hosts are on two different sub-networks, the packet is sent from the source host to a gateway. The accreditation range of the destination and of the first hop gateway is checked at the source when selecting a route. The gateway forwards the packet to the network where the destination host is connected. A packet may go through a number of gateways before reaching the destination.

On Trusted Solaris gateways, accreditation range checks are performed in certain cases. A Trusted Solaris computer routing a packet between two unlabeled hosts compares any IP label in the IP options portion of the packet against the accreditation range of the network interface. If no IP option is specified, the default label assigned to the sending host is compared to the accreditation range on the network interface through which the packet is going. Because the "write up read-down" MAC rule is enforced even on communications between unlabeled hosts, the default label of the sending host must be dominated by the default label of the destination host. In practice, two way communications would be impossible unless both unlabeled hosts shared a default label.

Each gateway maintains a list of routes to all destinations. Standard Solaris routing metrics allow routes to be chosen based on the shortest path to the destination. Extensions in Trusted Solaris 2.5.1 and later compatible releases enable trusted routing based on the shortest path to the destination that also satisfies security requirements. IP security options in a packet allow IP labels to be available for accreditation range checks on intermediate routers.

Trusted routing depends on all gateways recognizing extended RIP, the Routing Information Protocol. Therefore, trusted routing is only possible in an Intranet whose gateways are all known to use RIP, because routing in the Internet is done using other protocols.

Some sites using trusted routing need to enable communications with Trusted Solaris hosts that are on the other side of a cloud of unlabeled hosts when communications must go through one or more routers that do not understand labels. At these sites, the Security Administrator role needs to set up tunneling. (The terms *cloud* and *tunneling* are defined under “Setting Up Tunneling” on page 148.)

Choosing Routers

Because routes must be carefully chosen in the Trusted Solaris environment, the Security Administrator role needs to understand the security characteristics of all routers through which sensitive information is passing.

For the highest degree of trust, routes should be set up with Trusted Solaris computers as routers. If other types of routers are used, keep in mind that the Trusted Solaris security features are not always available on those routers, and without administrative action packets can be routed through routers without MAC security protection.

CIPSO and RIPS0 routers drop packets when they do not find the right type of information in the IP options section of the packet. For example, a CIPSO router drops a packet if it does not find a matching CIPSO label or a matching DOI in the packet's IP options section. Other types of routers not running Trusted Solaris software do not drop packets when they find labels they do not understand in the IP options section; they just pass the packets along. Be aware of these considerations when setting up communications between hosts, and make sure that packets are routed through the appropriate types of routers.

To support trusted routing, the Trusted Solaris routing tables are extended to include security information along with the metric for the number of hops to the destination, as described below.

Specifying the SRI

The set of security attributes necessary for trusted routing is called the *SRI* (for *security routing information*). The SRI always includes a minimum and a maximum label to establish the route's accreditation range:

As described on the `route(1M)` man page, the SRI can also incorporate other security attributes. The SRI is obtained from one of two possible sources:

- The dynamic routing software initially derives the SRI from the template assigned to the computer on the router.
- The Security Administrator role can enter the SRI manually in a static routing table.

Emetric

The *emetric* (Extended Metric) consists of both the standard routing metric and the SRI. The emetric is stored in each route's entry in the routing table. The routing software selects the shortest path that satisfies the security requirements by comparing emetrics. Alternately, the emetric can be entered manually for static routes using the `route(1M)`. (See "Routing Table" on page 141 for how routes are manually defined.)

If dynamic routing is used, the routing daemon, `in.routed` broadcasts a special type of security-enhanced response packet advertising the known routes.

Several routes through multiple gateways may exist between a sending and receiving host, and the emetric for each route may be different.

Routing Table

The routing table in the kernel of each host contains routes. Each entry in the routing table provides a route to a particular destination:

Destination	First hop gateway	Interface of gateway
(a specific host or network)	(first gateway in the route)	

The routing software tries to find a route to the destination host in the route tables. When the host is not explicitly named, the routing software looks for an entry for the (sub)network where the host resides. When neither the host nor the network where the host resides is defined, the host sends the packet to a default gateway, if one has been defined. Multiple default gateways can be defined, and each is treated equally. A pointer keeps track of which default gateway has been used most recently, and the next one in the list is used for the next routing.

Routing table entries are created either of the following two ways:

- **Dynamically** – The `routed(1M)` routing daemon dynamically creates the route entries including the metric.
- **Statically** – The administrator role creates static routes manually in one of two routing files. The administrator may or may not supply an metric with the route entry.

With a small network, it is feasible to set up routes manually, and to manually make changes to the routing table when conditions change. For example, many sites have a single gateway through which all communications go to the outside world. In these cases, the single gateway can be statically defined as the *default* on each host on the network. Manually configuring and maintaining static routes is less feasible with large networks.

Extended RIP

Xerox Routing Information Protocol (RIP) version is extended in the Trusted Solaris environment to supply security attributes along with a route's metric when the router advertises the route. The extended RIP is compatible only within an Intranet whose gateways all recognize RIP, because routing in the Internet is done using other protocols.

Determining Dynamic or Static Routing

The following figure shows how the presence or absence of certain files and programs on a Trusted Solaris host that is not a gateway determines whether static or dynamic routing is done.

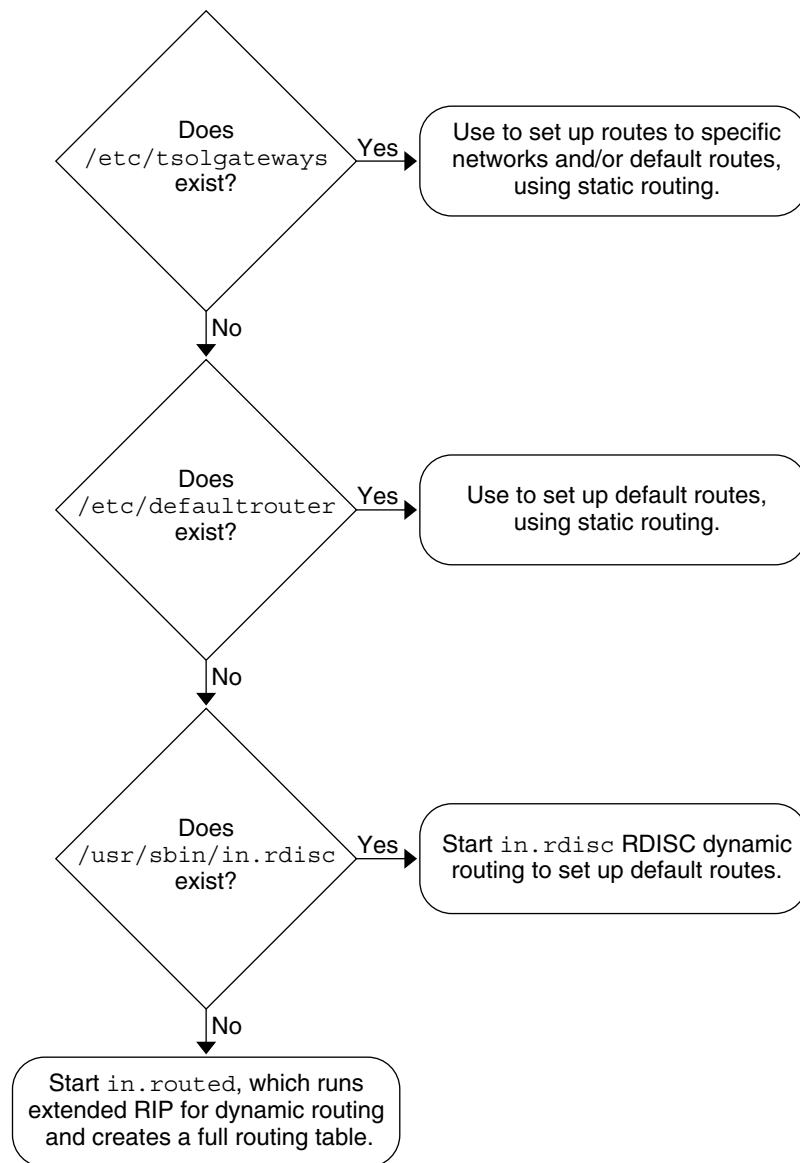


FIGURE 7-2 How a Host Determines Which Type of Routing to Do

Enabling a Single-Label Gateway to

Forward Packets at Multiple Labels

A single-label host (specified with a host type of Unlabeled or RIPS0) must be assigned a default label in its template. A minimum and a maximum label in the unlabeled host's template define an accreditation range that can be used for routing. Specifying the accreditation range enables a single-label gateway to be able to forward packets that it would not otherwise be allowed to receive based on its default label alone.

The trusted network software uses the accreditation range specified for a single-label gateway to decide which packets can be sent through that gateway. The packet being forwarded by the unlabeled gateway must be within the gateway's accreditation range.

Specifying Routing and Security for Remote Computers

This chapter provides implementation details and procedures for securing the network. This chapter includes the following procedures.

- “To Open the Security Families Tool” on page 148
- “To Construct Templates for Hosts” on page 149
- “To Assign Templates to Hosts” on page 149
- “To Replace the 0 . 0 . 0 . 0 Entry in the Local Tnrhdb File” on page 151
- “To Create a Wildcard Entry for Remote Hosts” on page 150
- “To Configure a Network Interface” on page 154
- “To Set Up Static Routes with Emetrics” on page 155
- “To Set Up Tunneling” on page 156

Assigning Security Attributes to Remote Hosts and Network Gateways

Each site’s Security Administrator decides which hosts should be allowed to communicate with the Trusted Solaris system and the security attributes of the hosts. The Security Administrator role uses the Security Families tool in the Solaris Management Console to assign security attributes to hosts by means of templates.

Templates can be assigned directly to a host or indirectly through a wildcard entry that assigns a template to a network address that includes the host. If a host does not have a template assigned either directly or indirectly, no communications can get through. Computers (hosts or routers) that share the same template are considered to be part of the same security family.

Optionally, the SMC Interface Manager tool can be used to assign security attributes to network interfaces, but doing so is useful only in limited circumstances when the defaults are not acceptable:

- To limit the range of labels at which communications are allowed through a network interface, the Security Administrator role can set a restricted label range. The default label range is ADMIN_LOW to ADMIN_HIGH.
- If it is desirable to be able to leave certain fields empty in a single template assigned to one computer or to a group of computers that is accessed through the same network interface, the Security Administrator can specify the values in an entry that applies to that network interface.

The entries assigned to network interfaces are looked at only if certain fields are left empty in the template assigned to a computer. If a value is not found either in the template that covers the host or in an entry that applies to the interface through which the remote computer is accessed, then a set of default values is applied.

Before assigning templates, the Security Administrator role should do the following:

- Review the existing templates.
 - Choose View->Details from the Security Families tool, which displays some of the values specified for each template.
 - Use the Security Families tool to bring up the Template Manager dialog box, select each template in turn and view its contents.
- Decide which templates should be used for each host and network.
- Modify existing templates or create any new templates needed for the site.

Setting Up Templates

Before assigning templates to hosts, have the following information available:

- A list of the available templates.
- A list of all the hosts and networks with which the hosts in the Trusted Solaris network are allowed to communicate.

Make the following decisions before starting:

- Decide which security attributes to apply to each host.
- Decide whether you can use existing templates or must modify them.

Storing Network Information

The Security Families tool in the Solaris Management Console stores template definitions in the `tnrhttp(4)` database and stores template to host assignments in the `tnrhdb(4)` database. The Interface Manager stores network interface definitions in the `tnidb(4)` file.

The Trusted Solaris version of the name service switch file, `nsswitch.conf(4)`, includes entries for `tnrhttp` and `tnrhdb`, which should be modified to suit each site's configuration. The default for NIS+ is shown below.

```
# TSOL
tnrhttp: files nisplus
tnrhdb: files nisplus
```

To modify these entries, the System Administrator role uses the Name Service Switch action. See “To Launch Local Administrative Actions” on page 32, if needed, for how to access the Name Service Switch action. To preserve the required file attributes (owner, group, mode and label), the role should not edit the `nsswitch.conf` file directly.

Modifying the Boot-Time Tnrhdb File

Local versions of the `tnidb(4)` and `tnrhdb(4)` files reside in the `/etc/security/tsol` directory on every Trusted Solaris computer. These local files are consulted before the system is configured and before the naming server is available. As delivered, the local `tnrhdb` file has a wildcard entry, `0.0.0.0:admin_low`.



Caution – The `admin_low` template may be a security risk on a Trusted Solaris network. Depending on site security requirements, the Security Administrator role may remove the `0.0.0.0` entry once the computer is installed. If it is removed, it must be replaced with entries for every computer the host contacts during boot. Alternatively, the `0.0.0.0` wildcard entry may be assigned a different unlabeled template.

See “To Replace the `0.0.0.0` Entry in the Local `Tnrhdb` File” on page 151 for how to change or remove the entry.

Setting Up Tunneling

Tunneling enables the sharing of emetrics for routes on an Intranet even when there is a non-Trusted Solaris cloud of hosts and gateways between two Trusted Solaris gateways. All hosts must be in the same Intranet with gateways using Trusted Solaris extended RIP. Without tunneling, the security response packets generated by extended RIP on one gateway cannot be received on the remote Trusted Solaris gateway to pass along the emetrics of its known routes.

To set up tunneling, the Security Administrator role creates a `tunnel` file on a Trusted Solaris gateway. The tunnel file contains the IP addresses of remote networks connected to Trusted Solaris gateways. Unlabeled broadcast packets containing security information are sent directly to the networks listed in the `tunnel` file, where they are picked by Trusted Solaris gateways. See “To Set Up Tunneling” on page 156.

Note – The term tunneling as used here has nothing to do with the IP-in-IP tunneling feature in the Solaris environment.

Managing Trusted Networking (Tasks)

▼ To Open the Security Families Tool

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.
2. Bring up the Solaris Management Console and load the Trusted Solaris Management Console in the appropriate name service scope.
3. Click the Trusted Solaris Configuration to open its list of tools.
4. Click the Computers and Networks tool and enter a password when prompted.
5. Double-click the Security Families tool.
All currently-defined templates display in the right hand pane. Use the online help.

▼ To Construct Templates for Hosts

1. **In the Security Administrator role, open the Security Families tool.**
See “To Open the Security Families Tool” on page 148 for the steps in detail.
2. **To modify an existing template, double-click the name of a template, then choose Action —> Properties.**
The Modify Template dialog displays with the name of the currently-selected template at its top.
3. **To add a new template, choose Action —> Add Template.**
Refer to the online help when adding a template.



Caution – When creating a new template, do not forget to change the Default Label. The Default Label in the default template is ADMIN_LOW, and normal users cannot work at that label.

4. **Supply the desired values in the tabs in the Template Manager.**
Refer to the online help for assistance.
5. **Click OK when done.**

▼ To Assign Templates to Hosts

1. **In the Security Administrator role, open the Security Families tool.**
See “To Open the Security Families Tool” on page 148 for the steps in detail.
2. **To change the assignment of a computer or network to a template, double-click the name of the ALL template.**
All computers and networks that are currently in the ALL family display in the right hand pane.
3. **Double-click the icon for the computer or network, then choose Action->Properties.**
The Modify Remote Host Entry dialog displays with the IP address of the network or computer at its top.
4. **Supply the desired values in the fields in the Template Manager, and click OK.**
5. **To assign an existing template to a computer or network, double-click the name of a template.**
All computers currently defined in the same Security Family display in the right hand pane.

6. **Choose Action->Add Host.**

The New Remote Host Entry dialog displays.

7. **Type in either the Hostname or the IP Address for any computer or network to which the template should be assigned.**

If a Hostname is entered, when you click the Load button the IP address is looked up. If an IP Address is entered, then the hostname is looked up. The IP Address field accepts any valid IPv4 or IPv6 address for the computer or network.

8. **Type in an optional Prefix Length that indicates the length of the network portion of the address.**

9. **Choose the name of a template from the Template pull-down menu.**

10. **Click OK.**

▼ To Create a Wildcard Entry for Remote Hosts

1. **In the Security Administrator role, open the Security Families tool.**

See “To Open the Security Families Tool” on page 148 for the steps in detail.

2. **Double-click the ALL template.**

3. **Choose Action —> Add host(s).**

4. **Click Wildcard, then give an IP address that ends in a zero (0).**

For example, 192.168.0.0 or 192.168.113.0.

5. **Assign an existing template to it.**



Caution – The wildcard entry allows any host on the wildcard’s network to communicate with this system at the label of the assigned template.

▼ To Change the tnd Polling Interval

By default, the tnd polls the trusted network databases every 2 minutes. The default for name service database polling is 30 minutes. You may want to change the tnd polling interval to match the name service interval once the network is up and running, and you have added all the templates and hosts.

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**

2. Open the Admin Editor from the `System_Admin` folder in the Application Manager, and edit the `/etc/init.d/inetsvc` file.
3. Find the 120 second polling interval, and change it to 1800 or another reasonable value.

▼ To Replace the 0 . 0 . 0 . 0 Entry in the Local Tnrhdb File

The local `tnrhdb(4)` file on each computer is used to contact the network at boot time. For greater security, you can remove the 0 . 0 . 0 . 0 wildcard entry. However, you must replace it with every remote address that the host contacts at boot time.

1. In the Security Administrator role, open the Security Families tool in the Files scope.
See “To Open the Security Families Tool” on page 148 for the steps in detail.
2. Double-click ALL, then select 0 . 0 . 0 . 0.
3. If you know all machines that this computer contacts, remove the wildcard entry by choosing Edit —> Delete.

4. To replace the wildcard entry, the following entries must be in the `/etc/hosts` or `/etc/inet/ipnodes` file, and in the `tnrhdb` database.

- An entry for this system, the name service master, and the loopback address, 127 . 0 . 0 . 1

The install team added these entries during configuration.

- An entry for every local IP address

The install team should have added these entries during configuration.

- One or more router entries

If the name service client is a router, list all the routers with which it needs to communicate during boot. Include broadcast addresses.

If the name service client is not a router, create a fallback network entry, such as 192 . 168 . 113 . 0.

- a. For a router, make the following entries by clicking Add —> Host(s).

Make sure all **network interfaces** are in the file. For example,

```
Host Name:  trusted-gw
IP Address: 192.168.112.111
Template:  tsol

Host Name:  trusted
IP Address: 192.168.113.111
```

Template: **tsol**

Make an entry for **every router** that this host communicates with. This is most easily done when the network uses static routing. For example,

Host Name: **gateway-2**
IP Address: **192.168.112.12**
Template: **unclassified**

Host Name: **gateway-3**
IP Address: **192.168.113.12**
Template: **unclassified**

Make an entry for **every broadcast and multicast address**. For example,

Host Name: **broadcast**
IP Address: **255.255.255.255**
Template: **admin_low**

Host Name: **multicast**
IP Address: **224.0.0.2**
Template: **admin_low**

Host Name: **broadcast-112**
IP Address: **192.168.112.255**
Template: **tsol**

Host Name: **broadcast-113**
IP Address: **192.168.113.255**
Template: **tsol**

The following shows the local `tnrhd` file with entries for a name service client with two interfaces. The client communicates with another network and routers.

192.168.112.111:tsol	<i>Interface 1 of this system</i>
192.168.113.111:tsol	<i>Interface 2</i>
192.168.113.5:tsol	<i>NIS+ master</i>
192.168.113.6:tsol	<i>Audit server</i>
192.168.113.8:tsol	<i>Mail server</i>
192.168.112.255:tsol	<i>Subnet broadcast address</i>
192.168.113.255:tsol	<i>Subnet broadcast address</i>
127.0.0.1:tsol	<i>Loopback address</i>
192.168.117.0:tsol	<i>Another Trusted Solaris network</i>
192.168.112.12:unclassified	<i>Specific network router</i>
192.168.113.12:unclassified	<i>Specific network router</i>
224.0.0.2:unclassified	<i>Multicast address</i>
255.255.255.255:admin_low	<i>Broadcast address</i>

- b. If the host being configured is not a router, click Add —> Host(s) to create a fallback entry so that the host can find its router.

For example,

Click the Wildcard button
IP Address: **192.168.113.0**
Template: **tsol**

▼ To Configure a Network Interface

1. **If adding a new interface, insert the network interface card, following the hardware and software installation steps in the guides shipped with the interface.**

The interface installation program installs a new device file called `hostname.device_abbreviation` in `/etc`.

2. **For a host with more than one network interface, do the configuration either for a router or multihomed host, as described in the the Solaris *TCP/IP and Data Communications Administration Guide*.**

3. **If the site security policy requires other than default settings for any interfaces, change the entries in the Interface Manager.**

As described in “Understanding Security Attributes Assigned to Network Interfaces” on page 134, interfaces on a computer running Trusted Solaris software are automatically detected by the trusted network software and assigned a default set of attributes. The Interface Manager shown below is used only when the security administrator role wants to change the defaults for an interface.

The Interface Manager tool is available when Scope=Files. The default attributes for network interfaces are shown in the following screen shot.

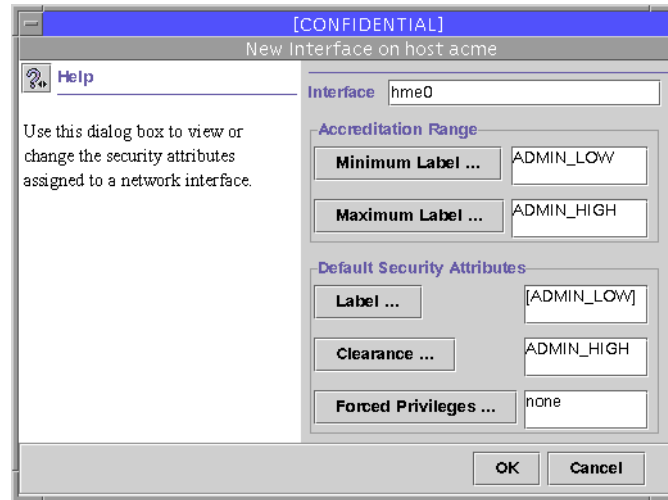


FIGURE 8-1 Interface Manager with Default Security Attributes

Note – Do not change the Min SL from ADMIN_LOW unless you have explicitly defined all routes.

▼ To Set Up Static Routes with Emetrics

1. Assume the System Administrator role and go to an ADMIN_LOW workspace.
2. Double-click the Set TSOL Gateways action in the System_Admin folder to open the /etc/tsolgateways file for editing.
See the tsolgateways(4) man page for more about the syntax and use of /etc/tsolgateways. The syntax of the emetric in tsolgateways is the same as for the route command.

3. Set up one or more default entries, if desired.

The first entry sets up a default route, using a specific gateway's address 192.168.113.36 and a metric of 1 to be used when there is no specific route defined for either the host or destination of a packet.

```
default 192.168.113.36 1
```

4. Set up one or more network entries, if desired.

The second line below shows a network entry set up with a standard metric. The third line shows a network entry set up with an emetric, setting a label range of PUBLIC to INTERNAL.

```
default 192.168.113.36 1
net 192.168.102.0 gateway-101 1
net 192.168.101.0 gateway-102 -m metric=2,min_sl="PUBLIC",
max_sl="INTERNAL"
```

5. Set up one or more host entries, if desired.

The new fourth line shows a host entry set up for a gateway host named trusted with an emetric setting a label range of PUBLIC to PUBLIC.

```
default 192.168.113.36 1
net 192.168.102.0 gateway-101 1
net 192.168.101.0 gateway-102 -m metric=2,min_sl="PUBLIC",
max_sl="INTERNAL"
host 192.168.101.3 trusted -m metric=2,min_sl="PUBLIC",
max_sl="PUBLIC"
```

6. Make sure there is an entry for any destination host(s) and gateway(s) in the local /etc/hosts file, or NIS+ hosts.org_dir table.

```
192.168.113.36 mynah
```

7. Make sure there is an entry for all destination hosts, network(s) and gateway(s) in the local /etc/security/tsol/tnrhdb file.

```
192.168.113.36:tsol1
```

8. Write and quit the file.

```
:wq
```

▼ To Set Up Tunneling

A forwarding host is any Trusted Solaris 8 4/01, Trusted Solaris 8, Trusted Solaris 7, or Trusted Solaris 2.5.1 gateway being set up to tunnel through one or more gateway(s) not running a Trusted Solaris 8 4/01, Trusted Solaris 8, Trusted Solaris 7, or Trusted Solaris 2.5.1 release to advertise the emetrics of its routes to the Trusted Solaris gateways on the other side.

- 1. Assume the Security Administrator role on the forwarding host and go to an ADMIN_LOW workspace.**
- 2. Use the Admin Editor action to create or open the /etc/security/tsol/tunnel file for editing.**

3. Enter one IP address of a target (sub)network on per line.

See the following example.

```
192.168.36.0
```

4. Write and quit the file.

```
:wq
```

5. To set up two-way routing using emetrics, repeat the previous steps on the remote gateway(s), specifying the IP address for the local network.

Managing Files and File Systems

This chapter describes how to manage files, directories, and file systems and how to share and mount files in the Trusted Solaris environment. This chapter contains the following procedures:

- “To Back Up Files ” on page 174
- “To Restore Files” on page 174
- “To Change Labels and Privileges With the File Manager” on page 175
- “To Set Security Attributes While Creating a Local File System” on page 176
- “To Set Security Attributes on a File System ” on page 177
- “To Specify Mount-time Security Attributes on the Command Line” on page 178
- “To Specify Mount-time Security Attributes in the `vfstab_adjunct` File” on page 178
- “To Share a Directory” on page 179
- “To Mount a TMPFS File System Using the Command Line ” on page 180
- “To Mount a CD-ROM with a HSFS File System” on page 180
- “To Automatically Launch a CD Player for an Audio CD-ROM” on page 180
- “To Listen to an Audio CD as any User or Role” on page 181
- “To Troubleshoot Mount Failures” on page 181

Requirements Unique to Trusted Solaris File Systems

The Trusted Solaris operating environment supports the same files and directories, most of the file system types, and all of the file system management commands in the Solaris operating environment. The Trusted Solaris environment adds security attributes. Whenever a file or directory is accessed, Solaris and Trusted Solaris security

attributes are checked for access decisions. The Trusted Solaris environment provides the following features and constraints on files and file systems:

- No order requirements are imposed on the labels of directories in a pathname.
- Files and directories can be created only at the same label as the containing directory.
- Privileged subjects can create files and directories and relabel existing files and directories at any valid label to create *upgraded* or *downgraded objects*.
See “To Change Labels and Privileges With the File Manager” on page 175.
- The system can be configured to show the names of upgraded files and directories. By default, their names are not visible.
To make the names are visible, the Security Administrator role changes the setting of the `tsol_hide_upgraded_names` switch in the `/etc/system` file as described in “To Change Configurable Kernel Switch Settings” on page 56.
- Directory names are cleared when a directory is removed. Clearing the names meets the object reuse requirement that the names of removed directories should no longer be accessible.
- Trusted Solaris symbolic links have labels.
- Multilevel directories (MLDs) appear in the file system as ordinary directories with a flag identifying them as MLDs.
- MLDs require no privilege to create, delete, or use.
- Read-down access to single-label directories (SLD)s within an MLD permits an unprivileged process to combine information from SLDs at its own and lower labels.
- If an MLD is mounted by a single-label computer, an SLD is mounted. The SLD corresponds to the label administratively assigned to the single-label computer in the trusted networking databases.

If, for example, a user’s home directory is automounted on an unlabeled computer, only the SLD with the default label assigned to the computer in the Security Families template is mounted. For example, if the default label for the computer is `INTERNAL_USE_ONLY`, then only the SLD at `INTERNAL_USE_ONLY` is mounted on the unlabeled computer.

Specifying Security Attributes on Files and File Systems

Security attributes can be specified at the level of an individual file or directory, or at the level of the file system.

If a needed attribute is not obtained elsewhere, a set of defaults is used. For rules about how attributes are obtained, see “Trusted Solaris Attribute Precedence Rules” on page 171.

Security Attributes on Files and Directories

The following attributes are present on objects in Solaris and Trusted Solaris file systems: User Id, Group Id, Permission Mode, and Access ACL (optional). Trusted Solaris files and directories have additional security attributes. The following table describes the extended security attributes provided in Trusted Solaris software.

TABLE 9-1 Trusted Solaris File and Directory Attributes

Extended Attributes	Description of Extended Trusted Solaris Attributes
Label	The label of the file or directory.
Forced Privileges	Optional. The set of privileges that an executable file is guaranteed to have available at start of execution. Must be a subset of the allowed privileges.
Allowed Privileges	Optional. The maximum set of privileges that an executable file is allowed to use during its execution. (Editing executable files causes them to lose all their privileges. Therefore, limiting the privileges that an executable can use to those in its allowed set provides a protection against Trojan Horses, since programs cannot use inheritable privileges if the programs have been edited.) Must be a superset of the forced privileges.
File Attribute Flag	Optional. The only supported file attribute flag is <code>public</code> . If the <code>public</code> flag is set, audit records are not generated when certain read operations are performed, even when these read operations are part of a preselected audit class, with one exception. If the audit pseudo-event for use of privilege (AUE_UPRIV) is included in a preselected audit class and if the operation involves the use of privilege, then an audit record is always generated.
Directory Attribute Flag	Optional. Flag indicating that a directory is an MLD

Specifying Security Attributes on Files and Directories

The Trusted Solaris File Manager enables users and administrators to change permissions on files and directories. It also enables authorized users and administrators to set privileges and labels on files and directories. Authorizations are required to change privileges and labels. Additional authorizations are required when the change is outside DAC or MAC policy.

Changing Labels and Privileges

The File Manager Selected menu has a Change Labels option to set the label. A user or role that has the `setLabel(1)` command in one of its profiles can also change labels. The File Manager Selected menu also has a Change Privileges option to set forced and allowed privileges on executable files. Changing forced and allowed privileges can also be done on the command line by any account that has the `setfpriv(1)` command in one of its profiles.

The following authorizations are required in order to set privileges and labels through the File Manager Selected menu options:

- Setting privileges requires the Set File Privileges authorization.
- Upgrading file and directory labels requires the Upgrade File Label authorization.
- Downgrading file and directory labels requires the Downgrade File Label authorization.

The following figure shows the File Manager Selected menu when the account has the required authorizations. See “To Change Labels and Privileges With the File Manager” on page 175 for how to change labels and privileges.

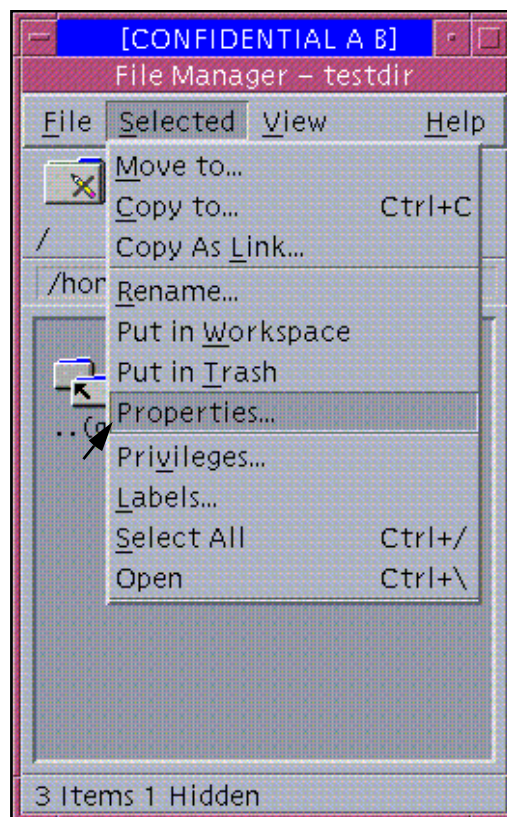


FIGURE 9-1 File Manager Selected Menu for an Authorized User

Changing File and Directory Attribute Flags

The `getfattrflag(1)` command gets the security attribute flags of a file or directory and the `setfattrflag(1)` command sets the public object flag on a file and sets the MLD flag on a directory.

Security Attributes on File Systems

File systems supported by Trusted Solaris software are characterized by whether their attributes can be changed or not. When the attributes can be changed, they are called *variable attribute* or *variable* file systems. File systems that do not support Trusted Solaris extended security attributes are called *fixed* because any attributes assigned to them (either at mount time or by default) cannot be altered.

Following are more details relevant for understanding and managing the various types of variable and fixed file system types:

- All `ufs`-type file systems are variable and therefore, all file systems installed with the Trusted Solaris software are variable.
For example, if you connect a hard disk containing an unlabeled file system directly to a Trusted Solaris computer, when the file system is `ufs`-mounted the unlabeled file system becomes a variable file system, with a default set of attributes shown in Table 9–2.
- An `nfs`-type file system mounted from a Trusted Solaris or TSIX NFS server is variable.
- An `nfs`-type file system mounted from an NFS server running another operating environment is fixed.
- `tmpfs` file systems are variable.
- These file system types are always fixed: `fdfs`, `hsfs`, `pcfs`.
- The `lofs`-type file system's attributes are those of the underlying file system. See "Mounting File Systems in the Trusted Solaris Environment" on page 167 for more information.

The following table shows the security attributes for variable-attribute file systems, with the default values that are used when none are specified.

TABLE 9–2 Variable File System Security Attributes with Defined Settings

Attribute	Description	Defaults
MLD prefix	The characters to use for the MLD prefix for MLDs on this file system	.MLD.
Label Range	The minimum and maximum sensitivity level for files and directories created on this file system	ADMIN_LOW to ADMIN_HIGH
Label	Label to infer for all files and directories on this file system that do not have an explicit label	None. NOTE: Files and directories in a fixed file system are assigned a default label when they are UFS-mounted, if the administrator has not assigned one.
Forced Privilege Set	Set of forced privileges to infer for all executable files on this file system that do not have explicit forced privileges	None.
Allowed Privilege Set	Set of allowed privileges to infer for all executable files on this file system that do not have explicit allowed privileges	None.

The Label Attribute

In variable file systems the label of each object is set when it is created and can be changed by an authorized user. In fixed file systems, a single label is assigned when the file system is mounted. The label can be changed only if an object is moved from the fixed file system. Because they are configured to have a single label when mounted on Trusted Solaris hosts, fixed attribute file systems are also referred to as *single-label file systems*.

The label is obtained differently when a fixed-attribute file system is NFS-mounted than when it is PCFS-mounted from a floppy disk or HSFS-mounted from a CDROM.

- An NFS-mounted file system is assigned the label that is specified in the Default Label setting in the Security Families template assigned to the remote computer from which the file system is NFS-mounted.
- For a PCFS- or HSFS-mounted fixed-attribute file system, the label is specified at mount time, either on the mount command line or in an entry in the `vfstab_adjunct(4)` file.

Specifying Security Attributes on Variable File Systems

(See “To Set Security Attributes on a File System ” on page 177. The Security Administrator role uses the `getfsattr(1M)` command to get the security attributes of a file system. The `setfsattr(1M)` command tunes the attributes set on an already-existing file system).



Caution – Do not change or explicitly set the security attributes of the `/`, `/usr`, or `/var` file systems on a Trusted Solaris host. The results are unpredictable.

Specifying Security Attributes on Fixed File Systems

When mounting a fixed-attribute file system, the Security Administrator role can specify security attributes on the command line with the `mount(1M)` command, in the `vfstab_adjunct(4)` file, or in the `/etc/auto_master` file other `autofs` maps (see `automount(1M)`).

Note – In the `mount` command, most of the keyword=value pairs used to specify security attributes with the `-S` can be specified with the `-o` option. If a keyword is followed by multiple values separated by commas, the keyword must be specified with the `-S` option because comma-separated values are not allowed after `-o`. Use of the `-o` option is preferable. For more about the security-related mount options that can be specified with the `-o` option, see “Mount Options Used for Protection” on page 168.

Any attributes specified at mount time are applied to all the files and directories in the mounted file system, if the files or directories themselves do not have the attribute. Any attributes on the file or directory are used. If the file or directory does not have an attribute and none is specified at mount-time, the defaults shown in Table 9–3 apply.

In fixed attribute file systems, the security attributes cannot change on an object as long as the object resides in the file system.

If, for example, the mounted file system `/spare` contains a file called `test`, no one can change the label of `/spare/test`. However, if `/spare/test` is copied into another directory such as `/tmp` or `/export/home/secadmin`, its label can be changed.

The following table shows the attributes that can be specified for a fixed attribute file system when the file system does not support the attribute, and the default values that apply if no value for the attribute is supplied.

TABLE 9–3 Attributes Assignable to Fixed File Systems

Attribute	-S or -o Option Keyword to Use When Mounting	Default Values
MLD prefix	<code>mld_prefix</code>	<code>.MLD.</code>
Label Range	<code>low_range</code> , <code>high_range</code>	<code>ADMIN_LOW</code> to <code>ADMIN_HIGH</code>
Label	<code>slabel=</code>	Mounted from a CD-ROM or floppy disk – the label of the mounting process Mounted from an NFS server – the default label of the server in the <code>tnrhdb</code> database
Forced Privilege Set	<code>forced=</code>	None
Allowed Privilege Set	<code>allowed=</code>	None

The following example shows a command line to NFS-mount a fixed attribute file system called `/spare` from an NFS server running the Solaris operating environment. The server is called `outside`. `/spare` is mounted with a label of `INTERNAL_USE_ONLY` using `mount` with the `-S` option on the command line as shown here:

```
$ mount -F nfs -S "slabel=INTERNAL_USE_ONLY" outside:/spare /spare
```

Mounting File Systems in the Trusted Solaris Environment

The Trusted Solaris `mount(1M)` command can be used to mount the types of file systems shown in the following table.

The table includes cross-references to `mount_*` mount man pages, when they are available for the named filesystem type, such as `mount_nfs(1M)` and `mount_ufs(1M)`. The mount man page describes security attributes that can be set for any file system type that supports using the `-S` option at mount time and describes the privileges, UID and GID that mount needs in order to succeed. The `mount_*` man pages give the subcommands that can be entered with the `-o` option for each filesystem type. See also “Security Attributes on File Systems” on page 163 and following for more about security attributes.

TABLE 9-4 Mount Types, Examples, and Notes

Type	When Used	Notes
FDFS	A pseudo file system type that allows a program to access its own file descriptors through the file name space.	MAC and DAC isolation are assured because each process can access only its own file descriptors. The mode (0666), group (root), and owner (root) are fabricated by the kernel and are not used in any DAC decisions. The label is of the backing file or directory. This is a fixed attribute file system.
HSFS	Mounts a file system from a CD device.	See <code>mount_hsfs(1M)</code> . In the Trusted Solaris environment, the file system can be given fixed attributes at mount time.
LOFS	A pseudo file system type that allows virtual file systems to be created that provide access to existing files using alternate pathnames.	See <code>lofs(7FS)</code> . In the Trusted Solaris environment, the security attributes are identical to those of the underlying file system.
NFS	Mounts a file system from a remote NFS server.	See <code>mount_nfs(1M)</code> . NFS mounts can be performed on fixed and variable attribute file systems.
PCFS	Mounts DOS file systems from a diskette.	See <code>mount_pcfs(1M)</code> and <code>pcfs(7FS)</code> . No extended attributes can be set on this file system type.

TABLE 9-4 Mount Types, Examples, and Notes (Continued)

Type	When Used	Notes
PROCFS	A pseudo file system provides access to the image of each process in the system. The name of each entry in the <code>/proc</code> directory is a decimal number corresponding to a process-ID. The owner of each “file” is determined by the process’s real user-ID.	In a Trusted Solaris environment, PROCFS is a variable attribute file system in which all the Trusted Solaris attributes are supported. Process access decisions are based on the DAC and MAC attributes of the <code>/proc</code> file, which are imputed from the underlying process’s DAC and MAC attributes. If the calling process has the <code>proc_owner</code> privilege, then the process can get information at the same label about processes not owned by the caller. If the calling process has <code>proc_mac_read</code> privilege, the process can get information about a process that is owned by the caller when the process’s label dominates that of the caller or is disjoint. The restrictions for modifying are more granular than the ones for reading. See the <code>proc(4)</code> man page.
TMPFS	Mounts in memory a temporary file system that uses swap pages, either in primary memory or on swap storage. The contents disappear at reboot.	Often <code>/tmp</code> is mounted as a <code>tmpfs</code> . The advantage is a huge increase in speed of access to whatever the temporary file system contains, since the information is retrieved from memory instead of from a disk. See <code>mount_tmpfs(1M)</code> .
UFS	Mounts a file system from a local disk.	See <code>mount_ufs(1M)</code> . UFS file systems can have fixed mount time attributes assigned or variable attributes assigned at creation or later. See “Specifying Security Attributes on Variable File Systems” on page 165.
AUTOFS	Automounting mounts file systems with the AUTOFS type.	See <code>automount(1M)</code> .

Note – The CACHEFS file system type is not supported.

Mount Options Used for Protection

The `mount(1M)` command can be used with the `-o` option followed by one of four protection options. The options are also valid in the `vfstab(4)` file. Some options can be used to protect the data on the file system being mounted, while others prevent a Trojan Horse attack initiated from the mounted file system. The mount restrictions shown in the following table are supported on all file system types. The Default Values column shows the values used when no option is specified.

TABLE 9-5 Mount Restrictions, Default Values

Description	Default Value	Alternate Value
Disallow write operations	<code>rw</code>	<code>ro</code>

TABLE 9-5 Mount Restrictions, Default Values *(Continued)*

Description	Default Value	Alternate Value
Ignore set user id bits on executables	suid	nosuid
Ignore forced privilege sets on executables	priv	nopriv
Disallow opens on device special files, preventing the use of devices from non-standard directory locations	devices	nodevices

Note – The `ro` and `suid` options to disallow writes and ignore set user ID bits are available in the Solaris version of the `mount` command.

Summary of Attributes on Various File System Types

The following table indicates how different file systems support the various file system attributes. See the key in Table 9-7.

TABLE 9-6 Attributes Supported by the Supported File System Types

Attribute	TNFS	UFS/TMPFS/SLNFS	PCFS/HSFS
Allowed privileges	FS	MT	MT
Forced privileges	FS	MT	MT
CMW label	FS	MT (label only)	MT (label only; from host's template)
MLD prefix	FS	MT	MT
Label range	FS	MT	MT
File system attribute flags	FS	none	none
Object attribute flags	FS	MT	MT
Mount flags	MT	MT	MT
Access ACL	OBJ	OBJ	none
File mode	OBJ	OBJ	*
File owner	OBJ	OBJ	*
File group	OBJ	OBJ	*

Type	Where Attribute Obtained
FS	From the file system
MT	From attributes specified at mount time
*	For HSFS with Rock Ridge extensions: same as the object

TABLE 9-7 KEY to the File System Attributes Table

UFS	A UFS file system on a Trusted Solaris host
TNFS	A TNFS file system from a Trusted Solaris or TSIX server
TMPFS	A TMPFS file system
SLNFS	A NFSv2 file system or a NFSv3 file system from a single-label/unlabeled server
PCFS	A PCFS file system
HSFS	A HSFS file system

MLDs are supported only by the following file system types:

- ufs (always variable)

- `nfs-variable`
(NFS file systems mounted from Trusted Solaris servers)
 - `lofs`, and
 - `tmpfs`
-

Trusted Solaris Attribute Precedence Rules

A file or directory's attributes take precedence over the attributes on the containing file system. Attributes specified at mount-time take precedence over filesystem attributes already in effect for a file system. Any attributes not obtainable at mount time or from the file system are assigned from the defaults.

The following figure illustrates the precedence rules.

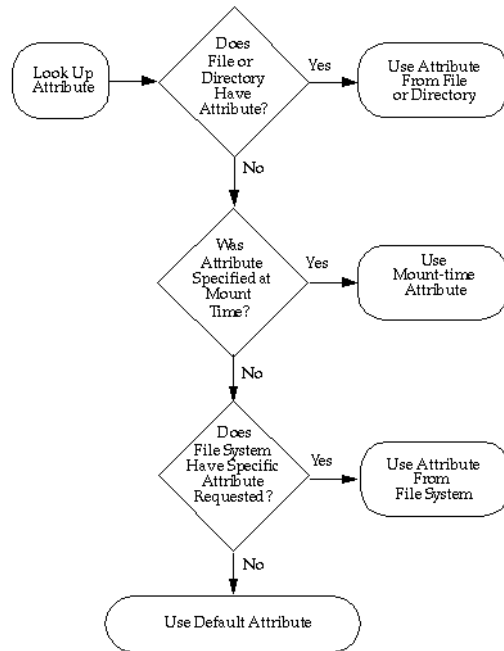


FIGURE 9-2 Trusted Solaris Attribute Precedence Rules

Trusted Solaris Software and NFS

Trusted Solaris software supports both NFS protocols supported in the Solaris operating environment and the Trusted Solaris 1.x release: NFS Version 2 (V2) and NFS Version 3 (V3) .

When a Solaris computer shares a file system using one of the NFS protocols above, the administrator of a computer running one of the following Trusted Solaris releases: 2.5.1, 7, 8, or 8 4/01, can specify the corresponding NFS protocol version to access the file system at a single label.

A Trusted Solaris computer can also specify the appropriate NFS protocol to share its own file systems with unlabeled client computers. A file or directory exported to an unlabeled client is *writable* if its label equals the label associated with the client

computer in its trusted networking database entries. A file or directory exported to an unlabeled client is *readable* only if its label is dominated by the label associated with the client computer.

Communications with computers running Trusted Solaris 1.1 and 1.2 releases is possible only at a single label. Both systems must assign each other a template with the unlabeled host type specified with the same single label.

Any file system being mounted from a NFS server running the Trusted Solaris environment must be mounted with *vers=2* and *proto=udp* mount options.

The NFS protocol used (whether it is NFS V2/V3, TNFS, TSIG/TNFS) is independent of the type of the local file system. Rather, the protocol depends on the type of the exporting computer's operating system. The file system type specified to the mount command or in the *vfstab* for remote file systems is always *nfs*.

Sharing Directories

Sharing directories for mounting by other computers works in the Trusted Solaris environment as it does in the Solaris environment. Trusted Solaris software provides two new Trusted Solaris mount options *nodevices* and *nopriv* to limit device use and privilege use. See "To Share a Directory" on page 179.

Troubleshooting Mount Failures

If an attempted mount fails, and if all the standard setup has been done as required in the base Solaris system (as described in "Mounting and Unmounting File Systems (Tasks)" in *System Administration Guide, Volume 1*), do the steps in "To Troubleshoot Mount Failures" on page 181.

Managing Files and File Systems (Tasks)

▼ To Back Up Files

1. Assume the Operator role or another role with the Media Backup rights profile.
2. Use one of the following backup methods:
 - `/usr/lib/fs/ufs/ufsdump` for major backups
 - `/usr/sbin/tar -T` with other options for small backups
 - A script calling either of the above commands

For example, the Budtool backup application calls the `ufsdump` command.



Caution – Only these commands preserve security attributes and can read multilevel and single-level directories correctly.

▼ To Restore Files

1. Assume the System Administrator role or any other role with the Media Restore rights profile.
2. Use one of the following methods:
 - `/usr/lib/fs/ufs/ufsrestore` for major restores
 - `/usr/sbin/tar -T` with other options for small restores
 - A script calling either of the above commands, such as Budtool.

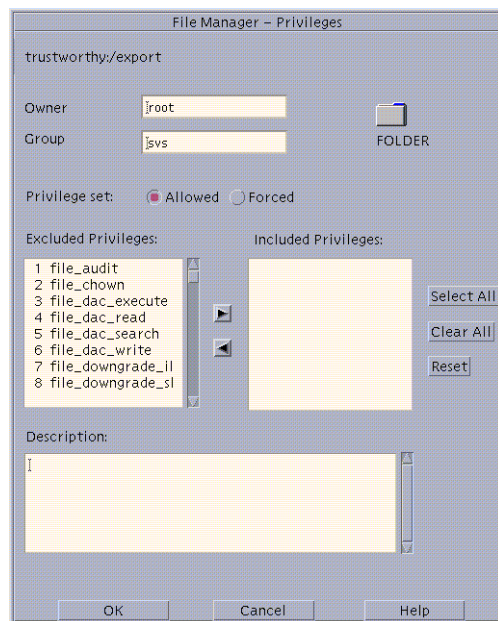


Caution – Only these commands preserve security attributes and can restore multilevel and single-level directories correctly.

▼ To Change Labels and Privileges With the File Manager

1. Assume the Security Administrator role and go to or create a workspace at the appropriate label.
2. Bring up the File Manager, navigate to the directory, and highlight the file whose privileges or label you wish to change.
3. To change privileges, choose Privileges from the Selected menu.

The File Manager Privileges dialog box displays as shown below.

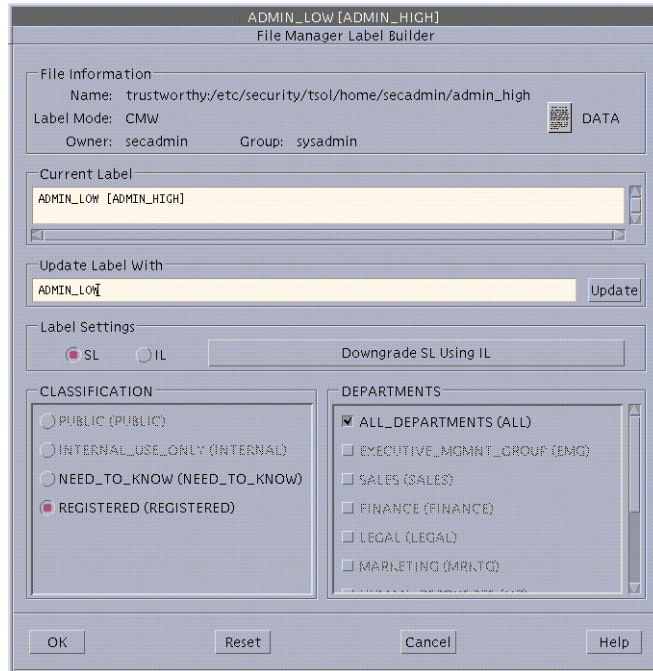


- a. On the File Manager Privileges dialog box, click the button for Allowed, and move the desired privileges from the Excluded to the Included list.
- b. Click OK, and repeat the above step for Forced Privileges.

c. Click OK.

4. To change labels, choose Labels from the Selected menu.

The File Manager Label Builder displays.



5. Enter a label by typing a label in the Update With field, or by clicking the desired label components.

6. Click OK.

▼ To Set Security Attributes While Creating a Local File System

1. Assume the System Administrator role and go to an ADMIN_HIGH workspace.

See “To Work at a Different Label” on page 33, if needed.

2. Using the File Manager or the `mkdir` command, make the mount point directory.

```
$ mkdir /newpublic
```


3. Use the Set Mount Points Action to edit the `/etc/vfstab` file with the following entry:

```
/dev/dsk/c0t3d0s3 /dev/rdisk/c0t3d0s3 /newpublic ufs 2 yes -
```

4. Write and quit the file.
5. Assume the Security Administrator role and go to an ADMIN_LOW workspace.

6. Execute the `newsecfs` command with the options that specify the desired alternative security attributes, then mount the file system.

The following example sets a label range of SECRET to SECRET.

```
$ newsecfs -l "Secret;Secret" /newpublic
$ mount /spublic
```

See the `newsecfs(1M)` man page for details.

▼ To Set Security Attributes on a File System

1. Assume the System Administrator role and go to an ADMIN_LOW workspace.
2. Use the Set Mount Points action to open the `/etc/vfstab` file and make sure that an entry exists for the file system:

```
/dev/dsk/c0t3d0s4 /dev/rdisk/c0t3d0s4 /spublic ufs 2 yes -
```

3. Change to an ADMIN_HIGH workspace.

See “To Work at a Different Label” on page 33 for changing the label of your workspace.

4. Enter the `umount` command to unmount the file system.

```
$ umount /spublic
```

5. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
6. Enter the `setfsattr` command with the appropriate arguments, then remount the file system.

The following example sets a label range of SECRET to SECRET.

```
$ setfsattr -l "Secret;Secret" /public
$ mount /spublic
```



Caution – Do not use proprietary names for mounted file systems. The names of mounted file systems are visible to every user.

▼ To Specify Mount-time Security Attributes on the Command Line

The following procedure mounts a `tmpfs`-type file system, `swap`, on `/mnt` with all allowed and all forced privileges.

1. Assume the System Administrator role and go to an `ADMIN_LOW` workspace.
2. Enter the `mount` command, using the `-S` option followed by any security attributes that you wish to specify.

```
$ mount -F tmpfs -S "allowed=all;forced=all" swap /mnt
```

▼ To Specify Mount-time Security Attributes in the `vfstab_adjunct` File

1. Assume the administrator role and go to an `ADMIN_HIGH` workspace.
See “To Log In and Assume a Role” on page 25, if needed.
2. Use the Set Mount Points action to open the `vfstab(4)` file for editing.
3. Specify the mount point as described in the `vfstab` man page and add filesystem-specific security options in the mount options column as desired.
See the filesystem-specific options in the `mount_*` man page for the file system type.
The example below shows a filesystem type of `ufs`, mounted with the Trusted Solaris `nodevices` and `nopriv` mount options and the Solaris `nosuid` mount option.

```
/dev/dsk/c0t3d0s4 /dev/rdsk/c0t3d0s4 /spublc ufs 2 yes nodevices,nopriv,nosuid
```

4. Save and close the file.

```
:wq
```
5. Assume the Security Administrator role and go to an `ADMIN_HIGH` workspace.
6. Use the Set Mount Attributes action to open the `vfstab_adjunct(4)` file for editing.

7. Copy and paste the template entry at the top of the file, and modify the copy.

```
#<mount point>; \  
#slabel=; \  
#forced=; allowed=; \  
#low_range=; hi_range=; \  
#mld_prefix=;
```

The example below gives the following security attributes to /spublic: all files in the file system get an slabel (label) of SECRET A, all allowed privileges, and all the file-related privileges.

```
# Assigns the Secret A label and label range, all file-related  
# forced privileges and all allowed privileges to an unlabeled file system  
#  
/spublic;\  
slabel="Secret A";\  
forced=file_audit,file_chown,file_dac_execute,file_dac_read,\  
file_dac_search,file_dac_write,file_downgrade_sl,file_lock,\  
file_mac_read,file_mac_search,file_mac_write,file_owner,file_setdac,\  
file_setid,file_setpriv,file_upgrade_sl;\  
allowed=all;\  
low_range="Secret A";\  
hi_range="Secret A";
```

8. Save and close the file.

```
:wq
```

▼ To Share a Directory

1. Assume the System Administrator role in an ADMIN_LOW workspace and invoke the Solaris Management Console.
2. Under Trusted Solaris Management Console, click *this-host*: Scope=Files, Policy=TSOL, then Storage. Provide a password when prompted.
3. Double-click Mounts and Shares, double-click Share, then choose Add Shared Directory from the Action menu.
4. Enter the file system you want to share.
5. After adding the directory, modify its attributes by double-clicking it, then modifying its properties.

Refer to the online help to guide you.

The following dfstab entry shares a book directory with the nodevices, nopriv, nosuid, and rw options.

```
share -F nfs -o nodevices,nopriv,nosuid,rw -d "Books" /spare/books
```

6. Click OK when done.

The tool modifies the `dfstab(4)` file, runs the `shareall(1M)` command, and starts the NFS daemon.

7. To confirm that the file system is shared, enter the `share(1M)` command with no options.

```
$ share
-                /spare/books    rw    "Books"
```

▼ To Mount a TMPFS File System Using the Command Line

1. Assume the system administrator role, and go to an `ADMIN_LOW` workspace.
2. In a profile shell, enter the `mount` command, using the `-S` option followed by any security attributes that you wish to specify.

```
$ mount -F tmpfs -S "allowed=all;forced=all" swap /mnt
```

The example mounts a tmpfs-type file system, `swap`, on `/mnt`.

▼ To Mount a CD-ROM with a HSFS File System

1. As any user or role, use the Device Allocation Manager to allocate the `cdrom_N` device.
2. If a CD in an allocated CD-ROM device contains a file system, the user is queried whether or not to mount the file system. Answer `yes` to mount the file system.

▼ To Automatically Launch a CD Player for an Audio CD-ROM

As described in Chapter 12, under “Mounting an Allocated CD-ROM Device” on page 214, if an allocated CD-ROM device contains an audio CD and if an audio action is specified in `rmmount.conf`, the audio action executes.

1. Assume the Security Administrator role in an `ADMIN_LOW` workspace.
2. Use the Admin Editor action to open the `/etc/rmmount.conf` file for editing.

3. **Add an action to automatically launch a CD player.**

The following example shows how the Security Administrator role could make an action in `rmmount.conf` for a CD player called `workman` installed in `/usr/bin`.

```
action cdrom action_workman.so /usr/bin/workman
```

4. **Save and close the file.**

```
$ :wq
```

▼ To Listen to an Audio CD as any User or Role

1. **Connect the speakers to the CD-ROM device and turn them on.**
2. **Complete the procedure “To Automatically Launch a CD Player for an Audio CD-ROM” on page 180.**
3. **Allocate the audio and the `cdrom_N` devices at your working label.**
4. **When prompted, insert the audio CD into the device.**

The specified CD player program is automatically launched.

▼ To Troubleshoot Mount Failures

1. **Make sure that the computer sharing the file system has been assigned a template on the Trusted Solaris computer doing the mounting.**

Use the Security Families tool in the Solaris Management Console to confirm that an appropriate template is assigned to an IP address that includes the NFS server. Look for the entry using the toolbox for the appropriate scope. If the NIS+ naming service is being used, bring up the SMC with the NIS+ scope. If NIS is being used, bring up the SMC with the NIS scope. If no naming service is being used, use the Files scope. See “Assigning Security Attributes to Remote Hosts and Network Gateways” on page 145 for more on how to assign templates to computers.

2. **If the computer is not running the Trusted Solaris operating environment, make sure the computer has been assigned a valid label in its template on the Trusted Solaris host.**

The label at which the host accesses the mounted directory must be the same as the label assigned in its template.

3. **Ensure that the mount is being done by the administrative role with the `mount` command in one of its rights profiles.**

In the default configuration, the Security Administrator role specifies the security attributes of mounts while the System Administrator role takes care of the Solaris

aspects of mounting.

4. **When mounting any file system from a NFS server running Trusted Solaris 1.x, make sure to use the `vers=2` and `proto=udp` options to the `mount` command.**

Managing Name Services

This chapter describes the differences in managing a name service in a Trusted Solaris environment. This chapter includes the following procedures:

- “To Enable Domain Administration from a Client” on page 185
- “To Save and Restore NIS Maps” on page 186
- “To Save and Restore NIS+ Tables” on page 186
- “To Use NIS and NIS+ Administrative Actions” on page 188

Managing Multiple Trusted Solaris Computers in a Security Domain

Setting up a name service master and clients (NIS and NIS+) is described in *Trusted Solaris Installation and Configuration*.

To achieve uniformity of user, host, and network attributes within a security domain with multiple Trusted Solaris computers, a naming service is used for distributing most configuration information. If a name service is not used, administrators should ensure that configuration information for users, hosts, and networks is identical in the local files on all hosts and any changes made on one host are made on all. See “Administering Remote Systems” on page 24, if needed.

A Trusted Solaris NIS or NIS+ master can manage data for Trusted Solaris and Solaris NIS or NIS+ clients.

A Trusted Solaris NIS+ master can also manage data for NIS clients (such as hosts running the Trusted Solaris 1.x operating environment) if NIS compatibility mode is used. NIS compatibility mode requires slightly different setup procedures than for a standard NIS+ server. NIS compatibility mode has security implications for NIS+

tables. For the differences and security implications, see “Using NIS-Compatibility Mode” in the *NIS+ Transition Guide*.

Trusted Solaris computers cannot be clients of Solaris NIS or NIS+ masters.

Managing Standalone Trusted Solaris Computers

Trusted Solaris computers may or may not be connected to a network with computers running other operating environments. A standalone Trusted Solaris computer may either be configured as its own name service master server or configured with no name service. If a Trusted Solaris standalone computer is configured without a name service, the configuration information is maintained in the `/etc`, `/etc/security`, and `/etc/security/tsol` directories. The administrative tools in the Trusted Solaris version of the Solaris Management Console enable the administrative role to specify Files scope so that the information is stored locally.

Enabling the root Role or a New Role to Administer a Name Server

If site security policy allows, root’s capabilities can be extended to allow the root role to do administration from a client, although this is not recommended.

For root to administer NIS+ from a NIS+ client, the name of the NIS+ client must be added to the NIS+ `admin` group using the `nisgrpadm(1)` command. If a new administrative role is created to administer NIS+ tables, an entry also must be added to the NIS+ `admin` group with the role’s principal name. See “To Enable a Role to Administer NIS+” on page 102 for an example.

Trusted Solaris NIS Maps and NIS+ Tables

Besides the standard databases listed in the “Information in NIS+ Tables” in *Solaris Naming Administration Guide*, Trusted Solaris software includes the following NIS maps/NIS+ tables: `tnrhdb(4)` and `tnrntp(4)`.

As in the Solaris operating environment, the administrator role can add NIS maps or NIS+ tables with protected data fields. As an administrative role, follow the procedures in the following books:

- *Solaris Naming Administration Guide*
- *Solaris Naming Setup and Configuration Guide*



Caution – Do not add new rows to the default NIS+ tables or modify the access rules defined for existing table fields.

Managing Name Services (Tasks)

▼ To Enable Domain Administration from a Client

The root role does this during initial configuration of the system, as described in “Configuring a NIS or NIS+ Client” in *Trusted Solaris Installation and Configuration*

1. Assume the System Administrator role and go to an **ADMIN_LOW** workspace.
2. Follow the procedures in “Connecting to the Name Server”.
3. **NIS+ ONLY:** For root to administer NIS+ from a NIS+ client, go to the NIS+ client and add the NIS+ client to the admin group by double-clicking the Add to NIS+ Administrative Group action and filling in the fields.

For example, the following two invocations of the Add to NIS+ Administrative Group action enable root to administer the NIS+ domain from the `good` and `good1` computers in the `security.example.com` domain.

```
Group Name: admin
Principal Name: good.security.example.com.

Group Name: admin
Principal Name: good1.security.example.com.
```

▼ To Save and Restore NIS Maps

Before installing a new Trusted Solaris release, you can save the information in your name service and restore it to the system after installation.

- **Use `ypcat(1)` to dump NIS maps into flat files and then propagate NIS maps from the files.**

See “Administering NIS” in *Solaris Naming Administration Guide* for how to propagate NIS maps from files.

▼ To Save and Restore NIS+ Tables

Before installing a new Trusted Solaris release, you can save the information in your name service and restore it to the system after installation.

1. **Create a script or use another means to dump the NIS+ tables into text files.**

Note – It is a good idea to dump the NIS+ tables into text files routinely, at least every time you make a change to NIS+.

- a. **To create a script, assume the security administrator role and use the Admin Editor action to create the script file at `ADMIN_LOW`.**

The following example shows a script called `nisscript` that the administrator role can create to do the dumps and to create a list of group members for later re-creation of the groups table.

```
#!/bin/sh
# nisscript
# nisplus tables into text files
#

mkdir -p /var/nis-backup
chmod 700 /var/nis-backup
cp /etc/.rootkey /var/nis-backup/dot-rootkey

# standard Solaris and Trusted Solaris tables
# NOTE: Add any tables created at your site

cd /var/nis/data
```

```

for i in audit_user auth_attr aliases bootparams ethers \
exec_attr group hosts netgroup netmasks networks passwd \
prof_attr protocols rpc services timezone tnrhdb tnrhnp \
user_attr shadow
do echo $i
/usr/lib/nis/nisaddent -d $i >/var/nis-backup/$i
done

# Use the following if you have any key value tables

for i in sendmailvars tntime
do echo $i
/usr/lib/nis/nisaddent -d -t $i.org_dir key-value >/var/nis-backup/$i
done

# get a list of each group and list each member in each group

mkdir -p /var/nis-backup/groups.list
chmod 700 /var/nis-backup/groups.list
for i in `nisls groups_dir | grep -v `:```
do nisgrpadm -l $i >> /var/nis-backup/groups.list/group.members
done

```

b. Assume the root role and run the nisscript created in the previous step at ADMIN_LOW.

2. For each group, execute the `nisgrpadm -l` command to list each of its members and save the output for use in step 7.

```
$ nisgrpadm -l group_name
```

3. Copy the directory containing the text dump files to a partition that you plan not to overwrite during installation or use `tar` to copy the files to tape or floppy.
4. After installation, if you did not save the text dump files in a saved partition, as root at ADMIN_LOW, create a staging directory for the text file dumps of NIS+ tables and restore the files from tape or floppy.

The screen example illustrates what to do when restoring the text NIS+ files to a `/setup/files` directory from a tape.

```

# cd /setup/files
# tar xv
bootparams
ethers
.
.
.

```

5. At the appropriate point in “Configuring the NIS+ Domain” in *Trusted Solaris Installation and Configuration*, re-create the NIS+ environment.

```
# nisserver -r -d domain-name.
```

Make sure to include the final period (.) in the domain's name.

6. In the Security Administrator role, at ADMIN_LOW, after running the `nisserver` command, run the `nispopulate` command in a profile shell with the `-F` and `-p` options followed by the name of the directory where the text dump files reside.

```
$ nispopulate -F -p /setup/files
```

7. Re-create the NIS+ groups and add members manually from the list of group members saved from the `nisscript` as described in step 2.

There is no easy way to recreate the NIS+ groups automatically.

▼ To Use NIS and NIS+ Administrative Actions

1. In an administrative role, open the System_Admin folder in the Application Manager.
2. To view the contents of tables or maps, use the actions View Table Contents or View NIS Map. Supply the table or map name when prompted.
3. To view the attributes of NIS+ tables, use the action View Table Attributes. Supply the table name when prompted.
4. To add a name service client, use the Create NIS+ Client or Create NIS Client actions.
5. To manage NIS+ administrative groups, use one of the following actions:
 - List Administrative Group
 - Add to NIS+ Administrative Group
 - Create NIS+ Administrative Group
 - Delete from NIS+ Administrative Group
 - Delete NIS+ Administrative Group

Managing Printing

This chapter describes how to set up labeled printing in the Trusted Solaris environment. This chapter contains the following procedures:

- “To Set Up Printing to a Non-Trusted Solaris Server ” on page 196
- “To Launch the Printer Administrator Action” on page 196
- “To Configure an Attached Printer” on page 196
- “To Configure a Network Printer for Labeled Output” on page 197
- “To Configure a Restricted Label Range for a Printer” on page 198
- “To Add Access to a Remote Printer” on page 200
- “To Enable Some Users to Print Without Banners and Trailer Pages ” on page 200
- “To Assign Printing-Related Authorization(s) to an Account” on page 201
- “To Suppress the Printing of Page Labels on All Print Jobs” on page 201
- “To Allow Some Users to Print Jobs Without Page Labels ” on page 201
- “To Set Up Public Print Jobs from an Unlabeled Print Server” on page 202

Requirements Unique to Trusted Solaris Printers

Solaris print utilities and databases have been modified to meet Trusted Solaris requirements for:

- Label-based control of access to printers and to information about queued print jobs
- Automatic printing of labels and other handling information on printer output and on mandatory banner and trailer pages

The System Administrator role manages printers. The Security Administrator role manages printer security, including the handlings of labeled output. The

administrators follow basic printer administration procedures described in the Solaris *System Administration Guide, Volume 2*. See especially the sections “Print Management (Overview)” and “Setting Up Printers (Tasks)”.

Configuring Printers in a Trusted Solaris Environment

The following table shows the tasks for configuring printers in a Trusted Solaris environment and the recommended roles and the tools that perform each task. The table provides links to procedures and other related documentation.

TABLE 11–1 Tasks for Configuring Printers

Role	Task	Tool	Notes
Rights Profile			
System Administrator Device Management	Configures printers	Printer Administrator action	See “To Configure an Attached Printer” on page 196, “To Configure a Network Printer for Labeled Output” on page 197, and “To Add Access to a Remote Printer” on page 200. See also “Starting Solaris Print Manager” and “Setting Up Printers (Tasks)” in the Solaris 8 <i>System Administration Guide, Volume 2</i> and following for how to do the configuration. Note – Where the instructions tell you to become superuser, do the steps at ADMIN_LOW in the System Administrator role.
Security Administrator Printer Security	Specifies a restricted label range for a printer (optional). The default is ADMIN_LOW to ADMIN_HIGH.	The Set Printer Label Range action or the add_allocatable(1M) command	See “To Configure a Restricted Label Range for a Printer” on page 198.

Printer clients can only submit print requests at labels that are allowed by the trusted network database entries for the printer client computer and printer server.

Allowing the Printing of PostScript Files

By default, users cannot print PostScript files. This restriction exists because a knowledgeable PostScript programmer could create a PostScript file that modifies the labels on the printer output.

If desired, the Security Administrator role can assign the Print PostScript authorization to trustworthy users and role accounts. The Security Administrator role should do so only if the account can be trusted not to spoof the labels on printer output and if permitting the printing of PostScript files is consistent with the site's security policy.

Adding Support for Additional File Types

A filter provided with the Trusted Solaris printing system converts text files to PostScript. Files converted to PostScript by any installed filter programs can be trusted to have authentic labels and banner and trailer page text because the filter's programs are trusted programs that are run by the printer daemon.

A site's System Administrator role can install additional filters, which then can be trusted to have authentic labels and banner and trailer pages. See the "Managing Character Sets, Filters, Forms, and Fonts (Tasks)" in *System Administration Guide, Volume 2* for how to add filters.

Setting Up Printers That do not Support Security Features

PostScript printers are the only types of printers that support labels and other handling information on printer output and on mandatory banner and trailer pages. The following types of printers function correctly, but they do not support page labels or labeled banner and trailer pages.

- Non-PostScript printers
- Printers connected to a print server that is not running the Trusted Solaris release
- Network printers that have not been configured from a Trusted Solaris computer
 - Jobs sent to a network printer print without labels and trailer pages if the network is not being managed by a Trusted Solaris print server. The network printer would have been configured in one of the two following ways:
 - Using the printer's own software supplied by the printer vendor to be a standalone node on the network
 - Using LP printer administration commands on a print server that is not running the Trusted Solaris release

If desired, the Trusted Solaris computer can be set up to send jobs to a printer connected to or managed by a computer (print server) that is not running Trusted Solaris software. Print servers connected to unlabeled servers can print jobs only at the single label that is specified for the print server in the trusted network databases on the Trusted Solaris computer. Jobs print without labels or trailer pages and without security information on banner pages.

Printing from unlabeled computers to a printer on a Trusted Solaris print server is supported.

Note – A user submitting a job from a single-label computer to a Trusted Solaris print server cannot cancel that job and cannot remove the job from the print queue. When a user sends a job from a labeled computer, the trusted network provides the UID of the user sending the print request. For unlabeled computers, the UID of the sender of the job is not available, so the UID assigned to the print job does not match that of the submitting user.

Managing Network Printers

Network printers can print labels on body pages and banner and trailer pages if the printer is managed by a Trusted Solaris computer. See “To Configure a Network Printer for Labeled Output” on page 197 for how to set this up.

Note – A network printer can print jobs only at the single label specified in the template that is assigned to the network printer’s IP address.

Controlling Whether Security Information is Printed on Print Jobs

The Security Administrator role can change the default for the printing of labels on body pages in the following ways:

- Give users an authorization on the print server to allow them to print jobs without labels on the body pages or print jobs without banner or trailer pages.
See “To Enable Some Users to Print Without Banners and Trailer Pages ” on page 200.
- Redefine fields in the `/usr/lib/lp/postscript/tsol_separator.ps` file on the print server in one of the following ways:

- Completely disable the printing of labels on body pages for all users, as described in “To Suppress the Printing of Page Labels on All Print Jobs” on page 201.
- Specify that another label or other wording is printed on body pages for all users.

By default, the Protect As classification is printed at the top and bottom of every body page. The “Protect As” classification is the dominant classification when the classification from the job’s label is compared to the minimum protect as classification that is defined in the `label_encodings` file.

The label printed at the top and bottom of banner and trailer pages as shown in the following figure is specified by means of the `/PageLabel` definition.

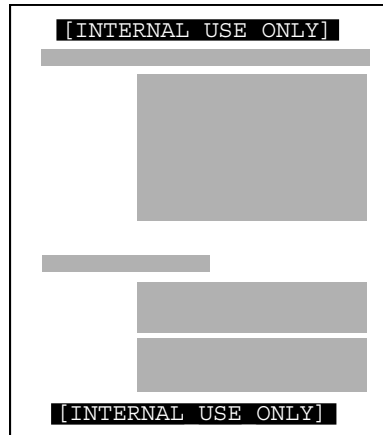


FIGURE 11-1 Job’s Label Printed on Body Pages

The `/HeadLabel` definition can be changed to put a different value or string at the top and bottom of the banner trailer pages or to print nothing at all.

Print Job Information on Banner and Trailer Pages

The following figures show a default banner page and the differences in the default trailer page. The names of the various sections are shown because they are needed when configuring what appears.

All the text and the labels and warnings that appear on print jobs are site-configurable. The text can also be replaced with text in another language for localization.

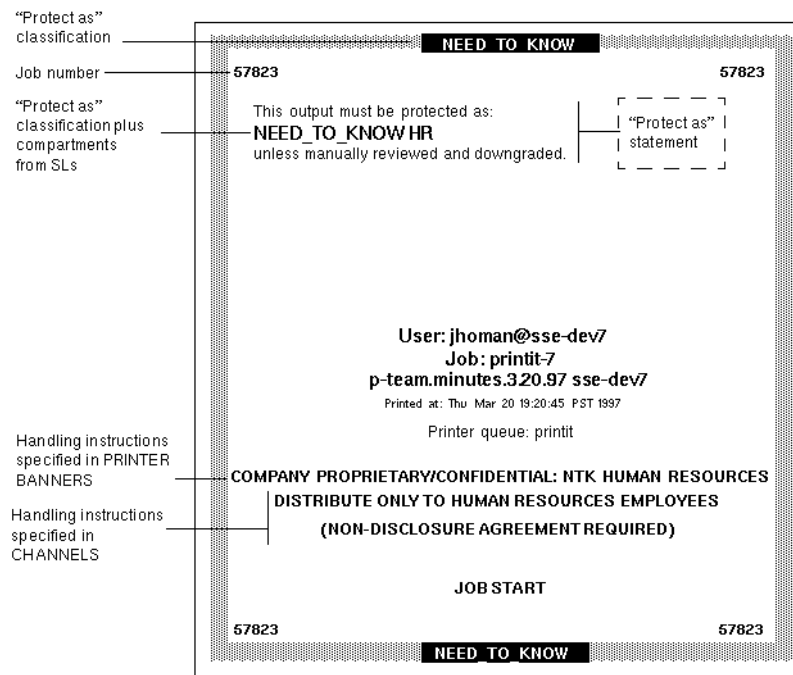


FIGURE 11-2 Typical Print Job Banner Page

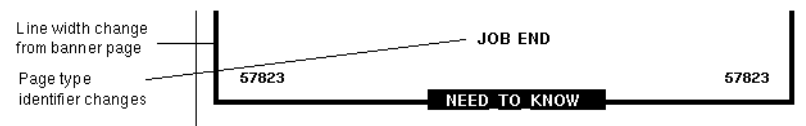


FIGURE 11-3 Differences on a Trailer Page

The following table shows aspects of trusted printing that the Security Administrator can change by assigning an authorization. For other printing-related authorizations see the *Trusted Solaris Administration Overview*.

TABLE 11-2 Modifiable Printing Features

What Can Be Changed	Authorization Name	How to Change
Whether individual users can print jobs without labels on body pages	Print without Label	Assign a rights profile with the Print without Label authorization to the user.

TABLE 11–2 Modifiable Printing Features *(Continued)*

What Can Be Changed	Authorization Name	How to Change
Whether all users can print jobs without labels on body pages	Print without Label	Enter AUTHS_GRANTED=solaris.print.unlabeled in <code>policy.conf</code> file.
Whether individual users can print jobs without banner or trailer pages	Print without Banner	Assign a rights profile with the Print without Banner authorization to the user.
Whether all users can print jobs without banner or trailer pages	Print without Banner	Security administrator enters Enter AUTHS_GRANTED=solaris.print.nobanner in <code>policy.conf</code> file.

The Security Administrator role can do the following to modify defaults that set labels and handling caveats on printer output:

- Localize or customize the text on the banner and trailer pages.
- Specify alternate labels to be printed in the various fields of the banner and trailer pages or at the top and bottom of body pages.
- Change or omit any of the text or labels.

Note – For how to do customizations or internationalization, see the comments in the `tsol_separator.ps` file.

Permitting Safe Jobs to Be Printed Without Labeled Pages

Certain users, such as technical writers, need to produce publicly-readable documents that do not have labels printed on the top and bottom of the pages. If a printer connected to a Solaris print server is available, the Security Administrator role can set up the users' environments so that the publicly-readable jobs go to the printer connected to the Solaris computer while jobs at all other labels go to Trusted Solaris computers. See: "To Set Up Public Print Jobs from an Unlabeled Print Server" on page 202. The procedure requires understanding of how to set up user accounts as described in Chapter 3, and computer network entries as described in Chapter 8.

Managing Printing (Tasks)

▼ To Set Up Printing to a Non-Trusted Solaris Server

Users send print jobs to the single-label printer at the same label assigned to the print server.

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.
2. Open the Solaris Management Console in the desired scope.
3. Click **Trusted Solaris Management Console**, then **Computers and Networks**. Provide a password when prompted.
4. Assign a template to the print server with the desired label.

The template is assigned to the IP address of the unlabeled print server.

See Chapter 8 for how the Security Administrator assigns a single label to an unlabeled computer.

▼ To Launch the Printer Administrator Action

1. Assume the System Administrator role and go to an `ADMIN_LOW` workspace.
2. In the `System_Admin` folder in the Application Manager, double-click the **Printer Administrator** action.
3. Choose `files` to update local files or choose either `NIS`, `NIS+(xfn)` or `NIS+` for a naming service.

▼ To Configure an Attached Printer

1. Connect the printer to a serial or parallel port on a print server using the appropriate cable, as described in the printer's installation guide.
2. Assume the System Administrator role on the print server, and go to an `ADMIN_LOW` workspace.

3. If the printer is connected to a serial port, make sure the correct baud rate is set, using the Serial Port tool from the Solaris Management Console Devices and Hardware manager.

See the printer documentation for the correct baud rate. See also “Adjusting Printer Port Characteristics” in *System Administration Guide, Volume 2*.

4. Bring up the Printer Administrator tool as described in “To Launch the Printer Administrator Action” on page 196.

5. Choose New Attached Printer from the Printer menu.

If needed, follow the procedure “How to Add a New Attached Printer With Solaris Print Manager” in the “Setting Up Printers (Tasks)” in *System Administration Guide, Volume 2*.



Caution – Do not change the Printer Type and File Contents settings from the default value of PostScript. If you do, printing will not work.

If the default printer label range of ADMIN_LOW to ADMIN_HIGH is acceptable, you are done.

6. To restrict the label range for the printer, go to “To Configure a Restricted Label Range for a Printer” on page 198.

▼ To Configure a Network Printer for Labeled Output

A network printer must be managed by a Trusted Solaris print server in order to print labeled output. A network printer prints only at a single-label assigned to it in a Security Families template.

1. Pick a printer name to be used as its host name, and assign the printer an IP address.
2. Set up the printer as described in the printer’s documentation.
3. Assume the System Administrator role on the Trusted Solaris print server, and go to an ADMIN_LOW workspace.
4. Add an entry for the printer using the Computers tool in the Solaris Management Console.

The scope of the toolbox that you load determines whether the entry is made in the local hosts file, NIS map or NIS+ table.

- a. Double-click Trusted Solaris Configuration->Computers and Networks->Computers.
 - b. Select Action->Add Computer.
 - c. On the Add Computer dialog, type the printer name in the Name field, type the printer's IP address in the IP Address field, and click OK.
5. Create a new unlabeled template assigning it the ADMIN_HIGH label.
- a. Double-click Trusted Solaris Configuration->Computers and Networks->Security Families.
 - b. In the Action menu, select Add->Template.
 - c. On the New Template dialog->Basic Information tab
 1. Assign a Name.
 2. Select Unlabeled from the Host Type menu and specify the Minimum Label and the Maximum Label as ADMIN_HIGH.
 3. Assign a Label and a Clearance of ADMIN_HIGH, and click OK in the New Template dialog box.
6. Assign the new template to the host name or IP address of the printer by double-clicking the icon for the new template.
7. In the Action menu, select Add->Host.
8. In the New Remote Host Entry dialog, enter the Host Name and IP address, then click OK.
9. Configure the printer on the Trusted Solaris computer using the LP administration commands.
- Complete the setup of the Network printer on the Trusted Solaris computer by following the procedure "How To Add A Network Printer Using LP Commands" in the "Setting Up Printers (Tasks)" in *System Administration Guide, Volume 2*.

▼ To Configure a Restricted Label Range for a Printer

Do this procedure only if you need to restrict the label range for a printer that is controlled by a Trusted Solaris print server. The default printer label range is ADMIN_LOW to ADMIN_HIGH.

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
See "To Log In and Assume a Role" on page 25, if needed.

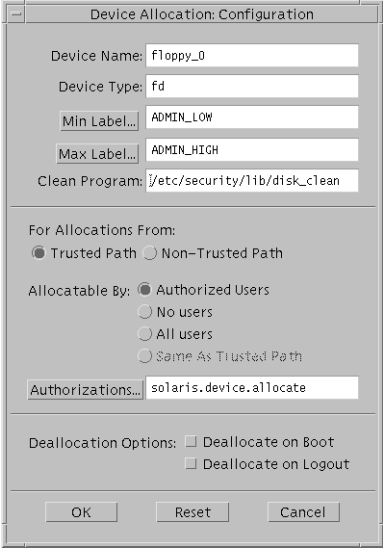
2. Bring up the Device Allocation Manager.

Either select the Allocate Device option from the Trusted Path menu or launch the Device Allocation Manager action from the Tools subpanel on the Front Panel.

3. Click the Device Administration button to display the Device Allocation: Administration dialog box.

4. Select the name of the new printer.

5. Click the Configure button to display the Device Allocation: Configuration dialog box, as shown in the following figure.



The image shows a dialog box titled "Device Allocation: Configuration". It contains several fields and options for configuring device allocation. The fields are: "Device Name" with the value "floppy_0", "Device Type" with the value "fd", "Min Label..." with the value "ADMIN_LOW", "Max Label..." with the value "ADMIN_HIGH", and "Clean Program" with the value "/etc/security/lib/disk_clean". Below these fields are two sections of radio buttons. The first section is "For Allocations From:" with "Trusted Path" selected and "Non-Trusted Path" unselected. The second section is "Allocatable By:" with "Authorized Users" selected, and "No users", "All users", and "Same As Trusted Path" unselected. Below the radio buttons is a text field for "Authorizations..." containing the value "solaris.device.allocate". At the bottom, there are "Deallocation Options:" with two checkboxes: "Deallocate on Boot" and "Deallocate on Logout", both of which are unselected. At the very bottom are three buttons: "OK", "Reset", and "Cancel".

6. Change the label range as desired by clicking the Min Label and Max Label buttons and using the label builders that display to select the desired label.

7. Click the OK button on the Configuration dialog box to save the label changes, click the OK button on the Administration dialog box to close it, and then close the Device Allocation Manager.

▼ To Add Access to a Remote Printer

Note – If either NIS+ or NIS was specified as the naming service when the print server is configured, this procedure is not needed on any NIS+ or NIS clients in the domain.


1. **On the local computer, access the Printer Administrator.**
See “To Launch the Printer Administrator Action” on page 196, if needed.
2. **See How to Add Printer Access With Solaris Print Manager in “Setting Up Printers (Tasks)” in *System Administration Guide, Volume 2*.**

▼ To Enable Some Users to Print Without Banners and Trailer Pages



Caution – If the Always Print Banner check box on the Printer Administrator dialog is checked, banner and trailer pages always print, even if the user has the `solaris.print.nobanner` authorization and uses the `-o nobanner` option to `lp`.

1. **Bring up the Printer Administrator on the print server.**
See “To Launch the Printer Administrator Action” on page 196, if needed.
2. **Make sure that the Always Print Banner check box is *not* checked.**



☐ Always Print Banner

3. **Exit the Printer Administrator.**
4. **Make sure that the `solaris.print.nobanner` authorization is in one of the profiles assigned to each user or role that is allowed to print without banner and trailer pages.**
See “To Assign Printing-Related Authorization(s) to an Account” on page 201, if needed.
5. **Instruct the user or role to submit jobs using the `lp` command with the option `-o nobanner`.**

```
trustworthy% lp -o nobanner staff.mtg.notes
```


▼ To Assign Printing-Related Authorization(s) to an Account

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.
2. Bring up the User Accounts tool.
3. Make sure that the desired print-related authorization is contained in one of the user's rights profiles.

▼ To Suppress the Printing of Page Labels on All Print Jobs

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.
2. Use the Admin Editor action to edit the `/usr/lib/lp/postscript/tsol_separator.ps` file.
See "To Edit a Local File" on page 33, if needed.

3. Find the following lines:

```
%% To eliminate page labels completely, change this line to
%% set the page label to an empty string: /PageLabel () def
/PageLabel Job_SL_Internal def
```

Note – The value of `Job_PageLabel` may have been changed at your site.

4. Replace the value of `/PageLabel` with an empty parentheses.

```
/PageLabel () def
```

▼ To Allow Some Users to Print Jobs Without Page Labels

1. Make sure that the Print Without Label authorization is in one of the profiles assigned to each user or role that is allowed to print jobs without labels at the top and bottom of each page.
See "To Assign Printing-Related Authorization(s) to an Account" on page 201, if needed.

2. **Make sure that the user or role submits jobs using `lp` with the option `-o nolabels`.**

```
trustworthy% lp -o nolabels staff.mtg.notes
```

Doing this procedure enables an authorized user or role to print jobs without labels when working at any label.

▼ To Set Up Public Print Jobs from an Unlabeled Print Server

Files that are available to the general public may be printed on an unlabeled printer.

1. **In the `tnrhd`/`tnrhtp` entries that define an unlabeled print server, assign to the print server the appropriate label.**

For example, a site may label files that are available to the general public as `PUBLIC` or `UNCLASSIFIED`.
2. **Do the following three steps for each user or role allowed to print publicly-readable files without page labels.**
 - a. **Make sure that the public label is in each account's personal label range.**
 - b. **Instruct each user to define the `PRINTER` variable in the appropriate shell initialization file in the user's publicly-labeled home directory SLD.**
 - i. **Go to the publicly-labeled home directory SLD.**
 - ii. **Open the `.login` or `.profile` file (as appropriate) for editing.**
 - iii. **Define the `PRINTER` variable to be the name of the printer connected to the unlabeled print server.**

When a printer named `nolabels` is connected to a single-label print server whose label is `PUBLIC`, the `.login` or `.profile` file in the `PUBLIC` SLD directory would have the following environment variable defined.

```
setenv PRINTER nolabels
```
 - iv. **Write and quit the file.**
 - c. **Have each affected account log out and log in again to put the changed printer definitions in effect.**
 - d. **Have each affected account create and print jobs that need to be printed without labels from within the publicly-labeled SLD.**

Managing Devices

This chapter describes how to protect information on devices. This chapter contains the following procedures:

- “To Save Files With Security Attributes to a Tape” on page 216
- “To Set or Modify Device Policy for a Device” on page 217
- “To Revoke or Reclaim a Device” on page 218
- “To Play an Audio CD” on page 218
- “To Add a Device” on page 218
- “To Configure a Serial Line for Logins” on page 220
- “To Assign Device Authorizations to an Account” on page 222
- “To Prevent File Manager Display After Device Allocation” on page 223
- “To Change or Add a Device Clean Script” on page 224

Controlling Access to Devices

The system administrator controls access to peripheral devices. Users can use a device only when the System Administrator role makes the device allocatable. Devices that the System Administrator makes nonallocatable cannot be used by anyone. Allocatable devices can be allocated only by authorized users. The Security Administrator role restricts the labels at which a device can be accessed.

Following are some highlights of device management in the Trusted Solaris environment:

- An unauthorized user in the default Trusted Solaris distributed system cannot allocate devices such as tape drives, CD-ROM drives, or floppy disk drives.
- A normal user with the Allocate Device authorization can import or export information at the label at which the user allocates the device.

- Users invoke the Device Allocation Manager to allocate devices when logged in directly. When logged in remotely, from scripts and from user-developed applications, the `allocate(1)` command is used.
- Only one authorized user at a time can access an allocatable device. After allocation, deallocating the device clears the device of data and frees it for allocation by another user.
- The label range of each device handled by the device allocation mechanism can be restricted by the Security Administrator. Normal users are limited to accessing devices whose label range includes the labels at which they are allowed to work. The default label range is `ADMIN_LOW` to `ADMIN_HIGH`.
- Nonallocatable devices are devices such as framebuffers and printers whose data is automatically cleared between users.
- Label ranges can be restricted for both allocatable and nonallocatable devices.

Setting a Label Range

To restrict direct login access through the console, the Security Administrator role can set a restricted label range on the framebuffer.

For example, a restricted label range might be specified to limit access to a publicly accessible computer. The label range enables users to access the computer only at a label within the framebuffer's label range.

When a host has a local printer, a restricted label range on the printer limits the jobs that it can print.

Managing Device Access Policies

In the Trusted Solaris operating environment, as in other UNIX systems, devices are represented by files called *device special files*. The discretionary access rules for devices are based on the same UNIX permission bits that apply to other types of files. The mandatory access rules that apply to devices are slightly different from those that apply to files or directories. The following table shows the default mandatory access control policy. These policies automatically apply to any new devices added to the system.

TABLE 12-1 Default Device Access Policy

Policy Type	Description	Default Policy
<code>data_mac_policy</code>	Label required to access the device	For reads and writes, the process' label must equal the device's label.
<code>attr_mac_policy</code>	Label required to access the device's attributes (by <code>acl(2)</code> , <code>chmod(2)</code> , <code>chown(2)</code> , and <code>stat(2)</code>)	For read access to the device's attributes, the process' label must dominate the device's label. For write access to the device's attributes, the process' label must equal the device's label.
<code>open_priv</code>	Privilege required to open the device	No privileges are required.
<code>str_type</code>	Only for STREAMS devices, specifies how the kernel stream head should control STREAMS messages	Device type stream. Unlabeled STREAMS message are allowed.

The Security Administrator role can change default policies and define new policies on each host by editing the `/etc/security/tsol/device_policy` file. Changes go into effect after a reboot. See the `device_policy(4)` man page for the keywords and values to use, and see also “To Set or Modify Device Policy for a Device” on page 217.

Initial Device Configuration Decisions

When configuring the Trusted Solaris environment on every system, the Security Administrator role sets device policy. After the system is up and running, the System Administrator role uses the Device Allocation Manager to add and configure devices, and to revoke an allocation, reclaim an allocated device from an allocate error state, or delete a device.

At system configuration, the Security Administrator needs to make the following decisions:

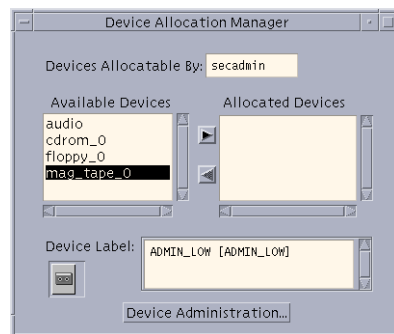
- Decide whether the default label range settings on nonallocatable devices are consistent with the site's security policy.
- Decide whether the default settings for the allocatable devices are consistent with the site's security policy.
- Decide whether to make additional devices allocatable.
- Decide which users, if any, should be allowed to allocate devices.
- Decide whether to use the default Allocate Device authorization or to create and require other authorizations for device allocation.

Decide whether to require separate conditions for a device to be allocated locally from the trusted path and for a device to be allocated without the trusted path either remotely or from a script. See the example of adding new device allocation authorizations in “To Add an Authorization to the Environment” on page 57.

Managing Devices

The `add_allocatable(1M)`, and `remove_allocatable(1M)` commands, the Add Allocatable Device action, and the Device Allocation Manager make changes to local versions of the `device_allocate(4)` and `device_maps(4)` files on the host on which they are run.

The following figure shows the Device Allocation Manager. The manager lists the allocatable devices currently present on the local system.



The Device Allocation Manager can be used only by users or roles that have the Allocate Device authorization.

The Device Administration button is visible to roles that have either one or both of the authorizations needed to administer devices, Configure Device Attributes, and Revoke or Reclaim Device.

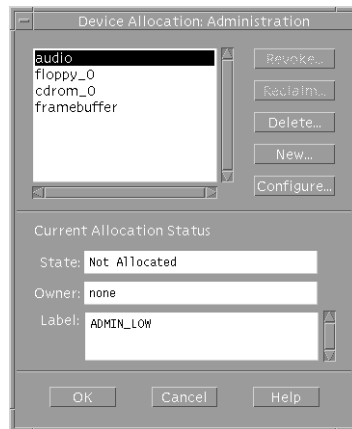
Making a Device Available

Users see an empty list when they are not authorized to allocate devices, or when the allocatable devices are currently allocated by another user or are in an error state. If a user cannot see a device in the Available Devices list, the user needs to contact the responsible administrator.

- If the user is not authorized but should be, the Security Administrator role can add the Allocate Device authorization to one of the user's profiles.
- If the device is not listed because it is already allocated or it is in an allocate error state, the administrator can force deallocation of a device or reclaim it from the error state.

Using the Device Allocation Manager

Clicking the Device Administration button launches the Device Allocation: Administration dialog box. This dialog box is used for reclaiming and revoking devices, deleting, or making entries for new devices.



Revoke – Click to force deallocation of the selected device.

Reclaim – Click to release the selected device from the allocate error state and leave it deallocated.

New and Configure – Click to create a new device or configure an existing device.

Configuring a Device

This section describes the information that can be specified for a device using the Device Allocation Configuration dialog box shown in the following figure.

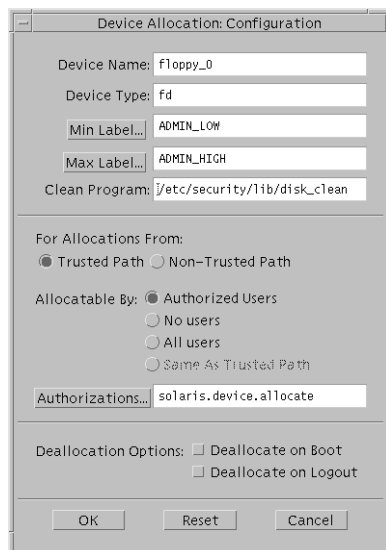


FIGURE 12-1 Device Allocation Configuration Dialog

Device Name and Device Type – Displays the name and device type. These fields can be edited when creating a new device.

Min Label and Max Label – Click to set the label range on the device. The default label range is ADMIN_LOW to ADMIN_HIGH. See “Initial Device Configuration Decisions” on page 205 for more about setting a device’s label range. These fields are valid for allocatable and nonallocatable devices.

Clean Program - Enter the path of a device_clean(1M) script for an allocatable device. If no device_clean script is specified at the time the device is created, the default is /bin/true. For how to write device clean scripts, see “Using Device-Clean Scripts” on page 211.

For Allocations From: Trusted Path or Non-Trusted Path – Click (Trusted Path) to require users to use the Device Allocation Manager when allocating the device. Click remote (Non-Trusted Path) to enable users to use the allocate command in a script or when remotely logged in to allocate the device.

By default, the Allocate Devices authorizations enables allocation from the trusted path and from outside the trusted path. Sites that are concerned about the potential risk of remote device allocation can restrict it. See “Authorizing Device Allocation” on page 210 for an example.

Allocatable By – Click one of Authorized Users, All Users, or No Users.

The No Users option is used most often for the framebuffer and printer, which do not have to be allocated to be used. But it is also used as shown in Table 12-3, to prevent an allocatable device from being accessed.

If no authorization is specified at the time the device is created, the default is All Users. If an authorization is specified, the default is Authorized Users.



Caution – Because the Add Allocatable action sets up a new device as allocatable by all users, the Security Administrator needs to click Allocatable By No Users when a device, such as the frame buffer and printers, should not be allocatable by anyone.

Authorizations – Click to change from the default authorization, `solaris.device.allocate`. See “To Add an Authorization to the Environment” on page 57 for an example of creating and adding new device authorizations.

Deallocation Options – Click Deallocate on Boot or Deallocate on Logout. to specify that any devices that are allocated by a directly-logged-in user are deallocated either at logout or at system boot or both.

Note – These options do not affect any devices allocated outside the trusted path (either during a remote login, or from a script or customer-written application) . Also, the `boot` command with the `-r` option can be used to force the deallocation of all devices at boot time.

Leaving devices allocated after logout could enable remote access to a device that otherwise can only be allocated locally. For example, a user could log in to one computer, allocate a device, then log out. The user then could log back in remotely to the first computer. During that remote session, the first computer’s microphone could transmit the talk around the first computer.

Handling of Allocated Devices at Boot

At boot time, by default, allocated devices are reallocated and remounted. The administrator can override the default at boot-time by entering the `device_clean(1M)boot` command with the `-r` option. To change the default permanently, the administrator checks the deallocation options in the Device Allocation Manager for every device that the administrator wants to Deallocate on Boot or Deallocate on Logout.

Authorizing Device Allocation

The Allocate Device authorization enables users to allocate a device and to specify the label to associate with information imported from it, or exported to it.

However, site security policy may require that you create separate authorizations for devices that are allocated from the trusted path and devices that are allocated without the trusted path. The following table shows an example:

TABLE 12-2 Requiring Separate Authorizations for Local and Remote Device Use

Device Name: cdrom_0
For Allocations From: Trusted Path
Allocatable By: Authorized Users
Authorizations: solaris.device.cdrom.local
For Allocations From: Non-Trusted Path
Allocatable By: Authorized Users
Authorizations: solaris.device.cdrom.remot

Alternatively, a site can allow a device to be allocatable only during local login sessions.

TABLE 12-3 Specifying Only Local Allocation of the Audio Device

Device Name: audio
For Allocations From: Trusted Path
Allocatable By: Authorized Users
Authorizations: solaris.device.allocate
For Allocations From: Non-Trusted Path
Allocatable By: No users

For added security around device allocation, the Security Administrator role can create a new allocate authorization, such as Allocate Remote Device. See “Adding New Authorizations” on page 48 and do “To Add an Authorization to the Environment” on page 57.

Enforcing Device Security

The security administrator decides who can allocate devices. The security administrator should make sure that any user who is authorized to use devices is trained and can be trusted to do the following:

- Properly label and handle any media containing exported sensitive information so that it does not become available to anyone who should not see it.

For example, if information at a label of `NEED TO KNOW ENGINEERING` is stored on a floppy disk, the person who exports the information must physically label the disk with the `NEED TO KNOW ENGINEERING` label and store the disk where it is accessible only to members of the engineering group with a need to know.

- Ensure that labels are properly maintained on any information being imported (read) from media on these devices.

An authorized user should allocate the device at the label that matches the label of the information being imported. For example, if a user allocates a floppy drive at `PUBLIC`, the user should only import information labeled `PUBLIC`.

The Security Administrator role also is responsible for enforcing proper compliance with the above-mentioned requirements.

Recovering From the Allocate Error State

As shown in Table 12–5, an allocatable device is in an *error state* if its ancillary file is owned by user `bin` and group `bin` with a device special file mode of `0100` and label of `ADMIN_HIGH`. One way that a device can be put into an `allocate error` state is by the `device_clean(1M)` scripts. A device-clean script puts a device into the `allocate error` state during deallocation until the user responds to prompts from the script and removable media is ejected. A role with the Reclaim or Revoke authorization can use the Device Allocation Manager to reclaim devices from the error state.

Using Device-Clean Scripts

A device-clean script is run any time a device is allocated or deallocated. The user who allocates the device usually deallocates it. If necessary, the Revoke button on the Device Allocation: Maintenance dialog box can be used by an authorized role to forcibly deallocate a device.

If your site adds additional allocatable devices to the system, the added devices may need new scripts. See the following descriptions of the existing device-clean scripts for ideas on how they work, and see also “Writing New Device-Clean Scripts” on page 213.

Device-Clean Script for Tape Devices

The `st_clean` device-clean script is used for all tape devices.

The `st_clean` script uses the `mt(1)` command with the `-rewoffl` option to clean the device. When the script is run during system boot, it queries the device to see if it is on line and has any storage media in it. If necessary, the script prompts the operator to eject the storage media, and then it displays the appropriate label for the user to write on a physical label on the storage media.

Until deallocation completes, 1/4 inch tape devices are placed in the `allocate` error state, and 1/2 inch tape devices are taken off line. The `allocate` error state forces an authorized user to manually clean up the device before a user can allocate it again.

Device-Clean Scripts for Floppy Disks and CD-ROM

The `disk_clean` script is used for both floppy disk drives and CD-ROM devices. When the `disk_clean` script is run during boot time, any media found in a device is ejected. Whether it is run at boot time or when the device is deallocated, if the `eject` succeeds, the script prompts the user to affix to the media a physical label with the appropriate label. If the `eject(1)` command fails, the device is put in the `allocate` error state.

When a file system from either a floppy or CD is mounted as part of allocation, a File Manager pops up with the current directory set to the mount point. The Security Administrator role can prevent the automatic display of the File Manager by following the procedure in “To Prevent File Manager Display After Device Allocation” on page 223. The mounting of file systems from floppy disks is handled differently from the mounting of file systems from CDs, as described in “Mounting an Allocated CD-ROM Device” on page 214 and “Mounting an Allocated Floppy Device” on page 214.

Device-Clean Script for Audio

The audiotool device is cleaned up using the `audio_clean` program.

This program performs an `AUDIO_DRAIN ioctl` to flush the device, and then an `AUDIO_SETINFO ioctl` to reset the device configuration to the default. In addition, this program retrieves the audio chip registers using the `AUDIOGETREG ioctl`, and any registers deviating from default are reset using `AUDIOSETREG ioctl`. Because the audio device does not contain any removable media, it does not require an external physical label, and therefore the label is not displayed by the `audio_clean` script.

Writing New Device-Clean Scripts

Devices that can be made allocatable include modems, terminals, and graphics tablets. The task of making any of these devices allocatable includes writing a new device-clean script. Device-clean scripts should also be created for any added tape devices, except for Xylogics or Archive tape drives, which can use the default `device_clean(1M)` script (`/etc/security/lib/st_clean`).

The default location for device-clean scripts is `/etc/security/lib`.

Device-clean scripts must return 0 for success and greater than 0 for failure.

Failure or inability to forcibly eject the medium must put the device in the `allocate` error state.

The `deallocate(1)` command passes four parameters to the device-clean scripts as shown here:

```
device_clean -[I|F|S] -[A|D] device_name label
```

The option letters `-I|F|S` help the script determine its running mode. `-I` is needed during system boot only. All output must go to the system console. `-F` is for forced clean up and `-S` is for standard cleanup. These are interactive and assume that the user is there to respond to prompts. With the `-F` option, the script must attempt to complete the cleanup if one part of the cleanup fails.

`[-[A] -[D]]` indicates whether the clean script is called from `allocate(1)` or `deallocate`.

The *device_name* field is a string with the name of the device.

The *label* field is a hexadecimal representation of the label.

Mounting an Allocated CD-ROM Device

When a CD-ROM device is allocated, the user is queried whether or not to mount the CD-ROM.

- If the CD contains a file system, answer `yes` to automatically mount the CD.
- If the CD is an audio CD, answer `no`.

When the answer is `no`, if an audio action is specified in `rmount.conf`, the audio action executes. By default, no audio action is specified.

To play an audio CD, the user must allocate both the audio and CD-ROM devices. The user can optionally manually invoke an audioplayer application after allocating the device. See “To Play an Audio CD” on page 218 for how the security administrator can set up an audio action for users.

Mounting an Allocated Floppy Device

File systems on floppy disks are not automatically mounted at allocation because the user may wish to create a new file system over an existing file system already on the floppy. Programs such as `fdformat(1)` or `newfs(1M)` can create a new file system only if the file system on the floppy device is not mounted. Therefore, before mounting an existing file system on a floppy, the `disk_clean` script asks the user whether or not to mount the file system.

If a floppy disk is not formatted, the `disk_clean` script asks the user whether or not to format the floppy.

After the file system on a floppy is mounted as part of device allocation, a File Manager pops up with the current directory set to the mount point.

Device-related Commands, Databases, and Files

See the man pages for the following commands and databases:

TABLE 12–4 Device-related Commands and Databases

Command or Database Name	Description
<code>allocate(1)</code>	Device allocation command line interface
<code>add_allocatable(1M)</code>	Add a device to <code>device_allocate(4)</code> , <code>device_maps(4)</code> , and create an ancillary file in <code>/etc/security/dev</code>
<code>deallocate(1)</code>	Device deallocation command line interface
<code>device_clean(1M)</code>	Device cleaning programs
<code>dminfo(1M)</code>	Report on specified device's entry in the <code>device_maps</code> file.
<code>list_devices(1)</code>	List devices specified in the <code>device_maps</code> file.
<code>remove_allocatable(1M)</code>	Remove a device from <code>device_allocate</code> , <code>device_maps</code> and delete its ancillary file from <code>/etc/security/dev</code> .
<code>device_allocate(4)</code>	Database for managing allocatable and some nonallocatable devices.
<code>device_maps(4)</code>	Database for device entries that are required for devices to be allocatable or to have their labels restricted.

Ancillary Files for Allocatable Devices

Each allocatable device has an ancillary file, which is a zero-length file in `/etc/security/dev`. The ancillary file is also referred to as a DAC file because the file must not only exist but its DAC permissions, owner, and group depend on its state.

The following table shows the DAC permissions, owner, and group for each of the possible states:

TABLE 12–5 Required Ancillary File Characteristics for Devices

Device State	DAC permissions (mode)	Owner	Group	Label
Allocatable	0000	bin	bin	ADMIN_LOW
Allocated	0600	<i>user</i>	<i>user's group</i>	<i>user's process's label</i>
Error State	0100	bin	bin	ADMIN_HIGH

Managing Devices (Tasks)

▼ To Save Files With Security Attributes to a Tape

This procedure can be done by any user or role that has the `tar` command in a profile.

1. Use the Device Allocation Manager to allocate a tape device.

The example allocates a device named `mag_tape_0`. See the *Trusted Solaris User's Guide* for more about how to allocate devices and specify the label at which the device is allocated.

2. Make sure the tape is physically labeled with the label of the current process, and insert the tape into the tape device when prompted.

The window in the example is titled Device Allocation for `mag_tape0` window.

```
st_clean: Insert tape into mag_tape0
```

```
st_clean: Make sure the tape is labeled CONFIDENTIAL
```

```
Press RETURN to quit window...
```

3. Enter the `tar` command with the `-T` security option.

```
trusted% tar cvT tartest
a tartest/(A) 1K
a tartest/ 0K
a tartest/file1(A) 1K
a tartest/file1 0K
a tartest/mld1/(A) 1K
a tartest/mld1/ 0K
a tartest/mld1/(A) 1K
a tartest/mld1/ 0K
a tartest/mld1/file50(A) 1K
a tartest/mld1/file50 1K
. . .
```

4. Use the Device Allocation Manager to deallocate the device.

Eject the tape from the device when prompted.

```
Please eject the tape in mag_tape_0
```

5. Make sure to protect the exported information at the security level on the media's physical label.

▼ To Set or Modify Device Policy for a Device

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
2. Determine the *driver_name* and *minor_name* and the device special file names for the device.
3. For an existing device, find the device name and minor name by doing a long listing of the device.

```
# ls -l /dev/dsk/c0t6d0s2
```

```
lrwxrwxrwx    1 root    root    51 Feb 29 1998 /dev/dsk/c0t6d0s2  
-> ../../devices/sbus@1f,0/SUNW,fas@e,8800000/sd@6,0:c
```

In the final element of the pathname, the string before the @ character is the driver name (sd in the example above) and the string after the colon is the minor name, (c in the example above).

4. For a new device, do the following.
 - a. Consult the hardware documentation for the device to obtain the device name and minor name and a list of all the physical device names.
See also, *Writing Device Drivers*.

- b. Create a new entry for the device in the `/etc/security/device_maps` file.

The name used for the device is arbitrary. In the third field, list all the physical device names for the device. The example shows all the physical and logical device names for the `cdrom_0` device.

```
cdrom_0:\  
sr:\  
/dev/sr0 /dev/rsr0 /dev/dsk/c0t6d0s0 /dev/dsk/c0t6d0s1  
/dev/dsk/c0t6d0s2 /dev/dsk/c0t6d0s3 /dev/dsk/c0t6d0s4  
/dev/dsk/c0t6d0s5 /dev/dsk/c0t6d0s6 /dev/dsk/c0t6d0s7  
/dev/rdisk/c0t6d0s0 /dev/rdisk/c0t6d0s1 /dev/rdisk/c0t6d0s2  
/dev/rdisk/c0t6d0s3 /dev/rdisk/c0t6d0s4 /dev/rdisk/c0t6d0s5  
/dev/rdisk/c0t6d0s6 /dev/rdisk/c0t6d0s7:\
```

5. Use the Admin Editor action to open the `/etc/security/tsol/device_policy` file for editing.
6. When the default policy for devices is not consistent with your site's security policy, create a specific entry or a wildcard entry for a new device or modify an existing entry for an already-specified device.

The default device policy is as shown in Table 12-1. For how to specify alternate policy settings, see the `device_policy(4)` man page.

7. Write the file and exit the editor.

▼ To Revoke or Reclaim a Device

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
2. Click the Device Allocation icon on the Tools subpanel.
3. Click the Device Administration button.
4. Check the status of a device by highlighting the name of the device and looking at the State: field.
5. If the State field is Allocate Error State, click the Reclaim button to correct the error state.
6. If a device is State is Allocated, do one of the following:
 - Contact the Owner to deallocate the device.
 - If the State field is Allocated, click the Revoke button to force deallocation of the device.
7. Click OK.

▼ To Play an Audio CD

The following procedure automatically launches a CD player. The user must have allocated both the audio and CD-ROM devices.

1. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
2. Open the Admin Editor from the System_Admin folder in the Application Manager to edit the `/etc/rmmount.conf` file.
3. Add your site's CD player program to the cdrom action in the file.

For example, at a site where workman CD program is installed, the following entry in `rmmount.conf` automatically executes `/usr/local/bin/workman` and launches the workman action.

```
action cdrom action_workman.so /usr/local/bin/workman
```

▼ To Add a Device

Follow the instructions in the *Installing Device Drivers* guide for the Solaris environment, if needed, then do the following Trusted Solaris-specific steps.

1. **If adding a new allocatable device, the System Administrator should create a `device_clean` script, if needed.**

A tape drive can use the default `st_clean` script as is, or the script can be modified to suit the site's security policy. Otherwise, a new `device_clean` script is needed. See "To Change or Add a Device Clean Script" on page 224 for the procedure.

2. **Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.**
3. **Click the Device Allocation icon on the Tools subpanel.**
4. **Click Device Administration, then click New....**
5. **Enter the Device Name and Device Type.**
6. **In the Device Map field, enter the pathnames for all the device special files associated with the device. Separate the pathnames with spaces.**
7. **(Optional) Set the label range on the device to be other than `ADMIN_LOW` to `ADMIN_HIGH`, by clicking the Min Label... and button and Max Label... buttons.**
8. **For Allocations From Trusted Path, choose an option from the Allocatable By: list:**

Authorized Users
No Users
All Users
Same as Trusted Path

Note – When configuring a printer, frame buffer, or other device that should not be allocatable, make sure to select No Users.

Same As Trusted Path applies only when Non-Trusted Path is selected.

9. **When you choose Allocatable by Authorized Users, the Authorizations field becomes active, and the `solaris.device.allocation` authorization name displays.**
If you have created site-specific device authorizations, enter them. See "To Add Site-Specific Authorizations to a Device" on page 220 for the procedure.
10. **Click Non-Trusted Path and click whether it should be treated the same as the Trusted Path.**
11. **If you choose Allocatable by Authorized Users, click the Authorizations... button to require site-specific authorizations to allocate the device from outside the trusted path.**
If you have created site-specific device authorizations, enter them. See "To Add Site-Specific Authorizations to a Device" on page 220 for the procedure.

12. Specify the Deallocation Options for the device when it is allocated locally through the trusted path.
13. Click OK to save your changes.

▼ To Add Site-Specific Authorizations to a Device

1. Assume the Security Administrator role and go to an **ADMIN_LOW** workspace or log in as a user who can assume a role with the **Configure Device Attributes** authorization.
2. Click the **Device Allocation** icon on the **Tools** subpanel.
3. Click **Device Administration**, select the device to allocate, and click **Configure....**
4. For **Allocations From Trusted Path**, choose **Authorized Users**.
When you choose Allocatable by Authorized Users, the Authorizations field becomes active, and the `solaris.device.allocation` authorization name displays.
5. If you have created site-specific device authorizations, click the **Authorizations...** button, and select the authorizations that the user must have to allocate the device.
6. Click **Non-Trusted Path** and click whether it should be treated the same as the **Trusted Path**.
Same As Trusted Path applies only when Non-Trusted Path is selected.
7. If you choose Allocatable by Authorized Users, click the **Authorizations...** button to add site-specific authorizations to allocate the device from outside the trusted path.
8. Click OK to save your changes.

▼ To Configure a Serial Line for Logins

1. Assume the Security Administrator role and go to an **ADMIN_LOW** workspace.
2. Bring up a SMC toolbox with the **Files** scope.

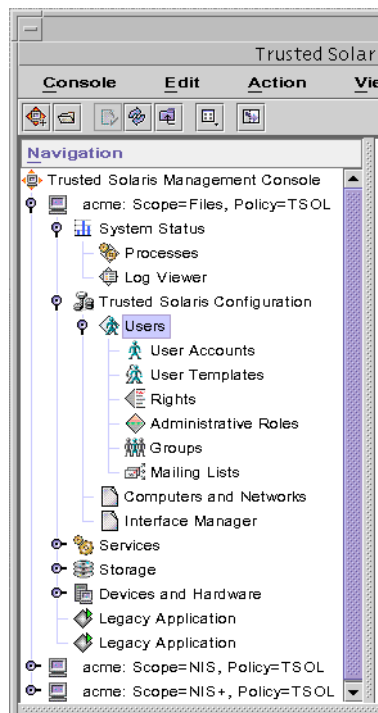


FIGURE 12–2 Solaris Management Console Tools

3. **Select Devices and Hardware, provide a password when prompted, and then double-click Serial Ports.**
Follow the online help for how to configure the serial port.
4. **Click the Device Allocation icon on the Tools subpanel on the Front Panel.**
The device's default label range is ADMIN_LOW to ADMIN_HIGH.
5. **To restrict the label range, click the Device Administration button, and then click New.**
 - a. **Enter /dev/term[a|b] for the Device Name.**
 - b. **Enter tty for the Device Type.**
 - c. **Enter /bin/true for the Clean Program.**
 - d. **Enter /dev/term[a|b] again for the Device Map.**
 - e. **Change the minimum and maximum labels if desired.**
 - f. **Choose No Users under Allocatable By.**

- g. Leave the Deallocation Options unset.
- 6. Click OK to save your changes.

▼ To Assign Device Authorizations to an Account

1. Assume the Security Administrator role, launch the Solaris Management Console in the appropriate scope, and click Users. Provide a password when prompted.
2. Double-click the User Accounts tool, and click the Rights tab.
3. Assign to the user a rights profile that contains the Allocate Device authorization.
If the defaults have not been modified, assign the rights profile Convenient Authorizations or All Authorizations.
4. To assign a rights profile to a role account, double-click the Administrative Roles tool, and double-click the role to be modified.
 - a. If the role should be able to allocate devices, choose a profile from the following table.

TABLE 12-6 Default Profiles that Include Device Allocation Authorization

Authorization Name	Default Profiles
Allocate Device	All Authorizations
	Convenient Authorizations
	Device Management
	Media Backup
	Media Restore
	Object Label Management
	Software Installation
	SSP Installation

- b. If the role should be able to revoke or reclaim devices, choose one of the following profiles.

TABLE 12-7 Default Profiles for Administering Devices

Name	Default Profile	Default Role
Revoke or Reclaim Devices	Device Management	secadmin
	All Authorizations	Not assigned

- c. If the role should be able to create or configure devices, choose one of the following profiles.

TABLE 12-8 Default Profiles for Creating Devices

Name	Default Profile	Default Role
Configure Device Attributes	Device Security	secadmin
	Host Alternate Pathing	secadmin
	All Authorizations	Not assigned

If none of the default profiles are appropriate for the account being reconfigured, the Security Administrator role can create a new profile that includes the device allocation authorization(s), either by themselves or along with any other commands needed by the profile's users to perform the desired work (such as the `allocate`, `deallocate`, and `tar` commands). Creating a new profile is described in "Adding or Modifying a Rights Profile" on page 87.

▼ To Prevent File Manager Display After Device Allocation

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.
2. Use the Admin Editor action to open the file `/etc/rmmount.conf` for editing.
3. Comment out the action for notifying the File Manager for the CD-ROM or floppy or both.

The example shows the `action_filemgr.so` commented out for both the `cdrom` and `floppy` devices.

```
# action cdrom action_filemgr.so
# action floppy action_filemgr.so
```

▼ To Change or Add a Device Clean Script

For background, see “Using Device-Clean Scripts” on page 211.

1. Assume the System Administrator role and go to an `ADMIN_LOW` workspace.
2. Use the Admin Editor to open a text file.
3. Write the script so that all usable data is purged from the physical device and that it returns 0 for success.
4. For devices with removable media, have the script attempt to eject the media if the user does not do so, and put the device into the allocate error state if the media is not ejected.
5. Copy the `ADMIN_LOW` script into `/etc/security/lib`.
6. Open the Device Allocation Manager from the Tools subpanel, and click the Device Administration button.
7. Highlight the name of the affected device and click the Configure... button.
8. Enter the name of the script in the Clean Program field.
9. Click OK until the Device Allocation Manager closes.

Adding Software

This chapter reviews how privileges are used by commands and actions. The chapter covers how to evaluate and install trustworthy software, including scripts, applications, and packages. This chapter contains these procedures:

- “To Mount a CD-ROM for Adding a Package” on page 245
- “To Give Forced Privileges to a Command” on page 246
- “To Create a New File Edit Action” on page 246
- “To Add Actions Outside of the System_Admin Folder” on page 248
- “To Make New Actions Available to the Rights Tool” on page 249
- “To Write a Profile Shell Script” on page 249
- “To Write a Standard Shell Script that Runs Privileged Commands” on page 251
- “To Save and Restore Privileges When Editing a File” on page 252
- “To Find Out Which Privileges a Program Needs” on page 252
- “To Make a Library Directory Trusted” on page 255
- “To Add Commands to the `/etc/inittab` File” on page 256
- “To Run `rc` Scripts With Security Attributes” on page 257
- “To Add Services to the `inetd.conf` File” on page 258
- “To Install a Java Jar File” on page 258

Types of Software

The following types of software can be added to the Trusted Solaris operating environment:

- Sun software products and third-party applications that neither understand nor enforce Trusted Solaris security policy
- New programs, created using Trusted Solaris programming interfaces, that understand labels and MAC (mandatory access control) and that work within

Trusted Solaris security policy

- New actions (created or approved by the Security Administrator role)
- Shell scripts (created or approved by the Security Administrator role)
- Additions to or modifications to commands that run during boot in run control scripts

Two distinct roles handle software evaluation and installation, the System Administrator role and the Security Administrator role.

Administrator Role Responsibilities

The System Administrator role installs software that meets the following criteria:

- Does not need to run with privilege
- Does not need to run with an effective UID or GID that differs from the real UID or GID of the invoking user
- Does not need to run at multiple labels
- Does not need to be added to a public directory

The System Administrator role also controls who can bring in software by granting or denying the device allocation authorization to individual users. An account with the device allocation authorization can import or export data at any single label within that user's clearance.

Security Administrator Role Responsibilities

The Security Administrator evaluates software for its ability to be trusted. As configured in the default system, the Security Administrator role can do the following:

- Import and export software at multiple labels
- Install software programs and CDE actions at `ADMIN_LOW` in the public directories (such as `/etc` and `/etc/dt/appconfig`) that allow use of the programs or actions by multiple users at all labels.
- Determine what privileges a program requires to succeed.
- Assign privileges to program files.
- Assign privileges that are in effect when a command or action is executed in a trusted process.

Note – Because applications and shell scripts, whether they are externally or internally obtained, are added to a site’s rights profiles as commands, the term *command* in this chapter refers to applications, site-developed executable programs, and shell scripts.

See “Assigning Privileges” on page 229 and the following sections, which define what it means for a program file to have privileges and for a command or action to inherit privileges.

See the *Trusted Solaris Developer’s Guide* for how programmers can manipulate privileges.

Privilege Enabling Mechanisms

The Trusted Solaris operating environment enables and enforces privilege inheritance by means of the system shell, profile shells, and the trusted window system. The software can also force privileges on executables with the `setfpriv` command. Privileged programs that require dynamically-loaded libraries search a trusted library list.

For a description of privileges themselves, see “Adding New Privileges” on page 49.

System Shell

The system shell, `sysh(1M)`, enables commands in *run control* (`rc`) scripts to execute with privilege. For every command in the `rc` script, the `sysh` consults a rights profile for security attributes. If no specific rights profile is listed for the command, the `/bin/sysh` shell consults the `boot` rights profile.

The `boot` and `inetd` rights profiles are local to each computer. They specify commands that require security attributes during booting. The `boot` profile specifies security attributes for commands in `/etc/init.d` and `/etc/inittab`. The `inetd` profile specifies security attributes for commands in `/etc/inetd.conf`.



Caution – Do not assign these profiles to any user or role. The results are unpredictable.

Profile Shells

Profile shells enable users and roles to execute commands with the security attributes that make the commands work. See the `pfexec(1)` man page for an explanation of the three profile shells, `/bin/pfsh`, `/bin/ksh`, and `/bin/sh`. All roles have a profile shell as their login shell. Users may or may not be assigned a profile shell, either as a login shell or as a shell made available in a rights profile.

Profile shells do not execute commands for roles unless the commands are issued within the trusted path.

Trusted Processes in the Window System

The following window system processes are trusted:

- Front Panel
- Subpanels of the Front Panel
- Workspace Menu
- File Manager
- Application Manager

The window system's trusted processes are available to everyone, but access to actions in the window system are restricted by an account's rights profiles. For example, the administrative actions that are in the `System_Admin` folder can only be used if they are in one of the account's profiles. Therefore by default, since the Check Encodings action is in the Object Label profile assigned to the Security Administrator role and the Set Mount Points action is not, the Security Administrator role can use the Check Encodings action but cannot use the Set Mount Points action.

In the File Manager, if an action is not in one of the account's profiles, the icon for the action is not visible. In the Workspace Menu, if an action is not in one of the account's profiles, the action is visible, but an error displays if the action is invoked.

The CDE window manager, `dtwm(1)` calls the `Xtsolusersession` script, which then works with the window manager to invoke actions launched from the window system. Just as the profile shell consults an account's rights profiles when the account attempts to invoke a command, `Xtsolusersession` also consults the account's

rights profiles when the account attempts to launch an action. In either case, if the action is in an assigned rights profile, the action is run with the security attributes specified in the profile.

Trusted Libraries

Dynamically-shared libraries used by `setuid`, `setgid`, and privileged programs can be loaded only from trusted directories. The list of trusted directories is in `/var/ld/ld.config`. Programs that require libraries from other directories will fail. See “Making Libraries Trusted ” on page 241 for how to extend the list of trusted libraries.

Assigning Privileges

Privileges can be forced on an object, or can be inherited by child processes. Therefore, the Security Administrator role has two ways to assign privilege:

1. By giving forced privileges to the executable file itself (for commands only).
2. By assigning inheritable privileges to a command or action in a rights profile.

When the command is executed in one of the shells that understands profiles (either the profile shells described in the `pfexec(1)` man page or the system shell, as described on the `sysh(1M)` man page, it executes with privilege. When an action is launched by a trusted process in the window system, it executes with privilege.

Giving Forced Privileges to an Executable File

The Security Administrator role can assign forced privileges to an executable file for a command by using the File Manager Privileges dialog box or by entering the `setfpriv(1)` command in a profile shell, as described under “To Give Forced Privileges to a Command” on page 246.

When a command with forced privileges is executed by any user in any shell, the forced privileges are put into the effective set of the executing program. The only way to prevent a user from executing such a command with privilege is to control access to the command itself. If you give the user only one profile shell to use, and do not assign the command, the user cannot execute the command.

To change the privileges on an executable file, the process’s label must allow MAC write access to the file. The process does not require DAC write permission. The default Security Administrator role can change the privileges on an executable file in an `admin_low` workspace. The forced and allowed privilege sets of a file can be changed by:

- The owner of the file or
- A process with the `file_setpriv` privilege or
- An account with the Set File Privileges authorization

Note – When you assign forced privileges using the File Manager Privileges dialog box, the software automatically assigns the same set of allowed privileges. However, the `setfpriv` command requires you to set the allowed and forced set appropriately in the same command line.

Assigning Inheritable Privileges to a Command or Action

After a site is configured, a privilege should be granted by a site's Security Administrator role only if the security administrator is convinced that the command or action will use the privilege in a trustworthy manner.

Privileges are available by inheritance when they are in a command or action's allowed privilege set. The Security Administrator role uses the Rights tool in the Solaris Management Console to specify inheritable privileges for a command or an action. The role then assigns the rights profile to a user or role, unless the profile is consulted by the system shell during boot. See "System Shell" on page 227 for the profiles that are not assigned to users or roles.

Passing Privileges to Child Processes

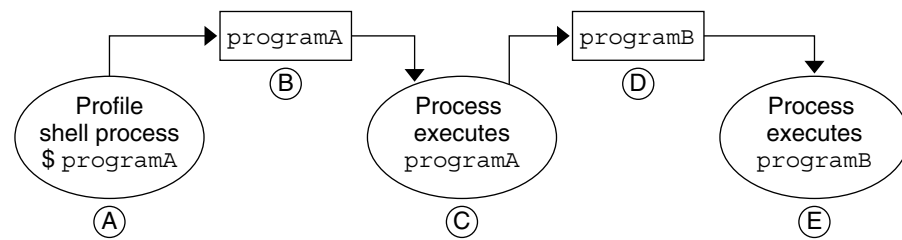
When a process executes a new program, the process's new inheritable set equals the process's old inheritable set before the new program was executed: `I[new]=I[old]`. The result is that the inheritable privileges available for one program to pass to another program are not affected by the forced or allowed privileges on the currently executing program.

The benefit of setting `I[new]=I[old]` without reference to allowed privileges is that privileges can be passed from a process executing a program that cannot use the privileges to one that can.

The benefit of setting `I[new]=I[old]` without reference to forced privileges is that forced privileges cannot be used by shell scripts.

Passing Privileges to Another Program

A process executing a program that has no allowed privileges cannot use any privileges because it cannot put any privileges into its effective set even if it inherits privileges from another trusted process. Such a process, however, can pass its inheritable privileges through to another program that it executes, one which might have allowed privileges and which therefore can use the inheritable privileges. The process executing the program without allowed privileges can pass privileges to another program because the inheritable set of the process is not affected by the lack of allowed privileges on the program. The following figure shows the inheritance mechanism.

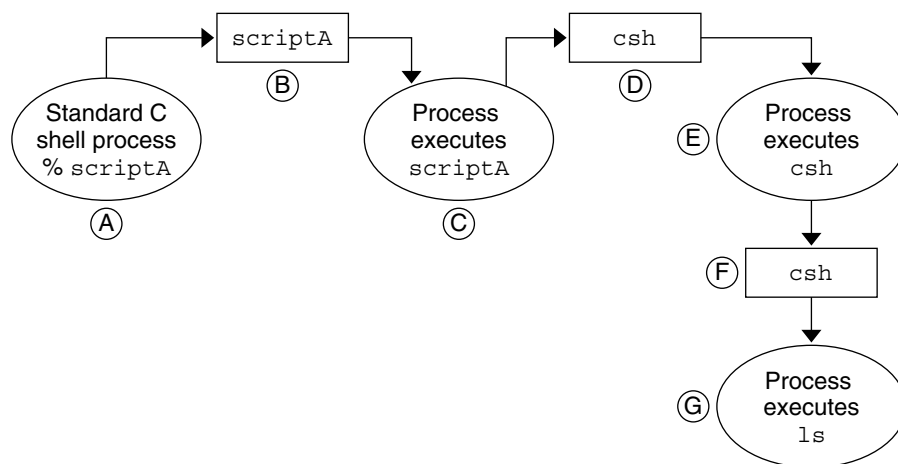


- (A) In one of the invoking account's profiles, `programA` has inheritable = 10, 12, 19, so `pfsh` sets its own inheritable = 10, 12, 19 and executes `programA`.
- (B) `programA` file's privilege sets: forced = none, allowed = none.
- (C) Process executing `programA` has inheritable = 10, 12, 19; effective = none.
- (D) `programB` file's privileges sets: forced = none; allowed = 10, 12, 19.
- (E) Process executing `programB` has inheritable = 10, 12, 19; effective = 10, 12, 19.

FIGURE 13-1 How an Unprivileged Program Can Pass On Privileges

Not Passing Forced Privileges via Shell Scripts

The inheritable set of a process cannot be increased by the forced privileges on the program. Any forced privileges on a shell script are not passed to commands invoked in a forced-privilege shell script. The result is that privileges cannot be used by shell scripts executed in standard UNIX shells, `sh(1)`, `csh(1)`, and `ksh(1)`. See the following figure.



- (A) The invoking shell `csh` for `scriptA` has no privileges, so its inheritable = none.
- (B) `scriptA`'s file privilege sets: forced = 10, 12, 19, allowed = 10, 12, 19.
- (C) Process executing `scriptA` has inheritable = none, effective = 10, 12, 19, permitted = 10, 12, 19.
- (D) `csh`'s file privileges sets: forced = none; allowed = all.
- (E) Process executing `csh` has inheritable = none, effective = none, permitted = none.
- (F) `ls`'s file privileges sets: forced = none; allowed = all.
- (G) Process executing `ls` has inheritable = none, effective = none, permitted = none.

FIGURE 13-2 How Forced Privilege Shell Scripts Are Prevented from Passing On Privileges

Creating and Using Shell Scripts

If an account has been assigned a normal UNIX shell (`sh`, `csh`, `ksh`), the account can create new shell scripts that can run any command in the system without privileges. Therefore, if none of its commands need privileges, a shell script can be used by anyone who has access to the script and its interpreting shell.

Making privileges available to commands that are invoked in shell scripts is done by the Security Administrator role.

Forced privilege commands are able to run with privilege in any shell because the forced privileges attached to the program file are available to the executing command. Assigning forced privileges to a `csh`, `sh`, or `ksh` shell script does not give any privileges to the commands executed by the shell script. Even though a shell that was

started from the script runs with the forced privileges, the shell does not have any privileges in its inheritable set. See the rules for how processes get privileges, which are described in “Passing Privileges to Child Processes” on page 230.



Caution – Shell scripts are vulnerable to being modified without detection. Before releasing shell scripts that use inheritable privileges, the security administrator should keep in mind that the protection against tampering that is available for programs is not available to shell scripts.

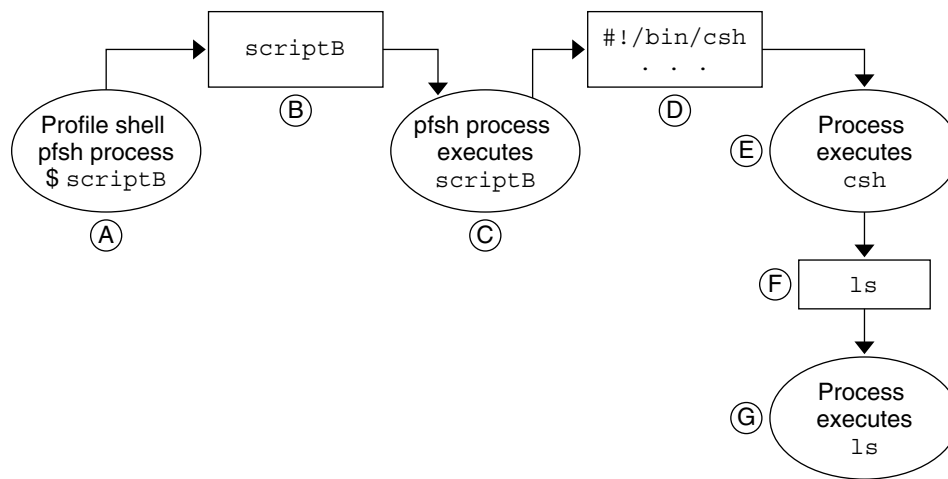
Summary of Shell Script Behavior in the Trusted Solaris Environment

- If none of its commands need privileges, a shell script can be created by anyone who is permitted to use a text editor.
- A shell script can be used by anyone who has access to the script and its interpreting shell.
- Forced privilege shell scripts do not pass privileges to commands that they contain.
- Allowed privileges on shell scripts have no effect on which privileges the programs executed by the shell script can use.

The allowed privilege set of the invoked shell's file is checked rather than that of the script's file.

- Standard shell scripts do not use the rights profile mechanism.
- A standard shell script that is invoked in a profile shell can pass privileges to commands that it runs if the Security Administrator role lists the shell script with any privileges required by its commands in one of the invoking account's profiles.

The shell script can pass any of its inheritable privileges to the commands it executes if the commands' executables have the allowed privileges. Because the commands are being run in a standard shell, it is no use to list them with privileges in one of the invoking account's profiles. The following figure shows a standard script executing in a profile shell.



- (A) In one of the invoking account's profiles, `scriptB` has `inheritable = 20, 22, 29`, so `pfsh` sets its own `inheritable = 20, 22, 29`.
- (B) `scriptB`'s file privilege sets: `forced = none`, `allowed = none` (but its file privilege sets are not consulted).
- (C) Process executing `scriptB` has `inheritable = 20, 22, 29`.
- (D) `csh`'s file privileges sets: `forced = none`; `allowed = all`.
- (E) Process executing `csh` has `inheritable = 20, 22, 29`; `effective = 20, 22, 29`; `permitted = 20, 22, 29`.
- (F) `ls`'s file privilege sets: `forced = none`, `allowed = all`.
- (G) `csh` process executing `ls` has `inheritable = 20, 22, 29`; `effective = 20, 22, 29`; `permitted = 20, 22, 29` so `ls` can do privileged operations.

FIGURE 13-3 How Shell Scripts Pass Inheritable Privileges Using a Profile Shell

- Profile shell scripts, that is, scripts that begin with the line `#!/bin/pfsh|ksh|sh`, can pass privileges to their commands in whatever shell they are running. The commands must be listed with the required privileges in one of the invoking account's profiles.

Using Profile Shell Scripts

Profile shell scripts behave differently when invoked by normal users than they do for administrative roles.

Profile Shells for Normal Users

- A shell script that invokes the profile shell can be executed by normal users on the command line in any shell.
- If the user has All Commands in a profile, the name of the profile shell script does not need to be explicitly added to any of the user's profiles.
- The commands in the profile shell script must be in one of the user's rights profiles, or the user needs the All Commands profile. Commands that need privilege must be assigned the required privileges in the profile.

Profile Shells for Administrative Roles

- A profile shell script (using `#!/bin/pfsh` or any other profile shell) must always be run in a profile shell.
- Roles cannot execute the profile shell from the command line or from a shell script (or bring up a GUI) without the trusted path.
- A role must have the name of any script using a profile shell *explicitly* listed in the Custom *role_name* Profile or another rights profile for the trusted path to be available. (For ease in troubleshooting, we recommend using the Custom *role_name* Profile for all customizations to a role's rights.)
- As is true for normal users, any commands in the profile shell script also need to be in one of the role's profiles.

See "To Write a Profile Shell Script" on page 249.

Editing Executables With Inheritable Privileges

To prevent unauthorized tampering with object code, any forced and allowed privileges previously given to a file are deleted whenever any executable program file is edited. This prevents someone from editing a file so that it uses privileges in a manner that was not originally intended. The Security Administrator role can save the list of privileges on such a file before editing it and restore them afterwards, as described in "To Save and Restore Privileges When Editing a File" on page 252.

Testing New Software for Security

The default Trusted Solaris programs and actions have already been assigned privileges, effective UIDs or effective GIDs when any of these attributes are required

for the programs or actions to do their work. This section outlines the issues and tasks associated with adding the following types of software:

- Sun software products and third-party applications that neither understand nor enforce Trusted Solaris security policy
- New programs, created using Trusted Solaris programming interfaces, that understand labels and MAC (mandatory access control) and that work within Trusted Solaris security policy
- New actions (created or approved by the Security Administrator role)
- Shell scripts (created or approved by the Security Administrator role)

Some programs run at a single level with no privileges required, so the Security Administrator role can simply install them at `ADMIN_LOW` in a public directory and assign them as desired as commands in the rights profiles of users and roles without assigning privileges or modifying any other attributes to make the programs work. Other programs that need to bypass security policy may need to be assigned privileges. The person who assesses the program needs to understand security and thoroughly understand what the new program is trying to accomplish.

Evaluating a Program for Security

The security administrator is responsible for testing and evaluating new software. When your site wants to add any existing programs to a Trusted Solaris environment, whether it is an application written outside of your organization, a Solaris software program, or a program written in house, the security administrator makes the final determination. Part of the determination is technical, and part is affected by site policy and procedures.

1. Find out if the application runs without changes in the Trusted Solaris environment.

If it runs without privilege or any modification, the System Administrator role can install the application.

2. If the program fails to run, find out why.

Some software packages and third-party applications written for the Solaris environment cannot run because of certain modifications made to the Trusted Solaris operating environment to enforce security policy. For example, software that links with the kernel may be incompatible with Trusted Solaris modified kernel data structures. For similar reasons, loadable device drivers and other software may not be capable of operating in the environment unless changes are made to the code.

If the program does not rely on aspects of the Solaris operating environment that have been modified for the Trusted Solaris environment, but it fails without privileges, find out what privileges or other attributes it needs.

3. If the program does require the use of privilege, assess whether the program will use its privileges in a trustworthy manner. See “Considering When to Add Privilege” on page 237.

If the program cannot use its privileges in a trustworthy manner and it cannot be modified, do not make it available.

4. If the program can safely run with the privileges or other security attributes in a manner that does not violate security policy, you may then assign the required privileges as described in “Assigning Inheritable Privileges to a Command or Action” on page 230.

If you make privileges available to a program, you need to make sure that any libraries used by the program are identified as trusted. See “Making Libraries Trusted ” on page 241.

5. If you can modify the program’s source code, a security consultant or programmer knowledgeable about security can modify the code.

These modifications might include privilege bracketing or adding code that makes the program aware of the Trusted Solaris security policy. You may then add privileges and trusted libraries. See “Making Libraries Trusted ” on page 241.

Considering When to Add Privilege

The most obvious type of program that fails without privileges is one that needs to run in the Solaris environment as root (such as a program that executes with `setuid` root). This kind of program can be assigned an effective UID of root in a rights profile that is assigned to an administrative role.

UNIX applications that need to violate DAC often are implemented to make careful checks before doing so on a user’s behalf. However, most applications are written in environments that do not have Trusted Solaris security mechanisms such as MAC. Therefore, a standard UNIX application does not carefully check MAC before violating MAC on a user’s behalf. An administrator who gives such a program a MAC-override privilege may unintentionally provide a way for users to override MAC arbitrarily.

Some of the security considerations to be assessed are illustrated by the behavior of `rcp(1)`, which is a commonly used UNIX program. The `rcp` command, which copies files across a network, runs with `setuid` root. Running as root allows the program to run with all privileges in a standard UNIX system. Although the program is allowed to bypass DAC restrictions, it checks for DAC permissions on a file to make sure the user who executed the has permission to access the file. But `rcp` has no knowledge of MAC restrictions. If you gave the `rcp` command the `file_mac_read` or `file_mac_write` privilege, it would not do the right kinds of checks for MAC relationships when accessing a file for a user; so `rcp` would not be able to use the privileges you assigned it in a manner that enforces the security policy of the system.

If you simply assign a similar program the privileges it needs to run and do not modify it to work within the security policy of the Trusted Solaris environment, the program violates system security. In order to make it run without violating system security, you would need to add to the program's source code. For example, if a program needed to bypass MAC restrictions when reading and writing files, you would need to modify the source code by adding the necessary MAC checks.

Some software may need privileges for reasons that are not obvious and sometimes not necessary for the program to succeed. Even if it is not performing any function that seems to violate system security policy, an application may be doing something internally that does violate security, such as keeping track of its operations in a shared log file, or reading from `/dev/kmem` (see `mem(7D)`). Sometimes these internal policy overrides are not particularly important to the application's correct operation but merely provide a convenient feature for users. If your organization has access to the source code, check if you can remove the operations that require policy overrides without affecting the performance of the application.

If the program would violate aspects of Trusted Solaris security policy, such as reading and writing files without doing MAC checks, then you should probably either make sure the required MAC checks are added to the source code, if you can, or not port the program.

Running a Program As Root

When an application has been written to run as root, the Security Administrator role has three options (all of which should be assessed for consistency with the site's security policy):

- If a *real* UID of root is not required, set up the application to run with an *effective* UID of root.
- Otherwise, set it up with a real UID of root.
- Find out what privileges the application needs and assign only the needed privileges, after determining that the application can use the privileges in a trustworthy manner.

Cooperating to Create a Trusted Program

Even though a program's developer can manipulate privilege sets in the program's source code, if the Security Administrator role does not assign the required privileges, the program will fail. The developer and security administrator cooperate when creating trusted programs.

Developer's Responsibilities

A developer who writes a program to be added to a Trusted Solaris environment must do the following:

1. Understand whether the program requires privileges to do its work.
2. Know and follow techniques, such as privilege bracketing, for safely using privileges in programs.
3. Be aware of the security implications when assigning privileges to a program and make sure that the program does not violate security policy.
4. Compile using shared libraries linked to the program from a trusted directory.

See the *Trusted Solaris Developer's Guide* for additional guidelines and examples of using privileges in programs.

Security Administrator Role's Responsibilities

The Security Administrator role must ensure that a program that uses Trusted Solaris system calls and routines not compromise the security of the Trusted Solaris environment in any way.

1. Make sure the programmer and the program distribution process is trusted.
2. From one of these sources, find out which privileges are required by the program:
 - a. Ask the programmer.
 - b. Search the source code for any privileges that the program expects to use.
 - c. Use the `runpd` command as described in "To Find Out Which Privileges a Program Needs" on page 252.
3. Scrutinize the source code to make sure it behaves in a trustworthy manner when using the privileges it needs to operate.

Adding Trusted Actions

The process of creating and using actions is pretty much the same in the Trusted Solaris environment as it is in a Solaris environment. Adding actions is described in the "Adding and Administering Applications" in *Solaris Common Desktop Environment: Advanced User's and System Administrator's Guide*.

In the Trusted Solaris environment, *use* of actions is controlled by the rights profile mechanism. Actions that are assigned security attributes in a rights profile can run with the assigned security attributes if they are invoked within one of the window system's trusted processes. In the Trusted Solaris environment, a number of actions

have been assigned security attributes in the rights profiles of administrative roles. The Security Administrator role can also use the Rights tool to assign security attributes to new actions.

The following table summarizes the main differences encountered in creating and using actions in the Trusted Solaris environment.

TABLE 13-1 Constraints on Actions in the Trusted Solaris Environment

Solaris CDE	Trusted Solaris CDE
New actions may be created by anyone within the originator's home directory, and a new action is automatically usable by its creator.	<p>An action is only usable by a user or role if the action is one of the account's rights profiles. The actions' search path has been changed so that actions in any individual's home directory are processed last instead of first. Therefore, no one can customize existing actions.</p> <p>If either the Create Action action or commands or actions that permit the editing of files are in an account's profile, the user or role <i>can</i> create a new action in the account's home directory, but the account may not be able to use the new action.</p> <p>There are two ways a user can use a new action: if the Security Administrator role adds the name of the new action to one of the account's rights profiles, or if the user has the All profile. The All profile turns off all checks for actions, and as a result any existing and potential actions may then be used by that account.</p> <p>If the account is allowed to use the action by its rights profiles, the account can launch the action from its home directory through the File Manager. The default System Administrator and administrator roles are permitted to place actions in public directories.</p>
Actions can be dragged and dropped to the Front Panel.	The Front Panel is part of the trusted path. The window manager recognizes only the administratively-added actions that are located in <code>/usr/dt</code> and <code>/etc/dt</code> subdirectories where system-wide action files are kept. Even if a normal user account or a non-administrative role account creates a new action in the account's home directory and has the All Accounts profile, new actions dragged to the Front Panel from the user's home directory are not recognized by the window manager, which only looks in the public directories.
The only way that actions can do privileged operations is if they are run by root.	If actions are specified to have privileges in one of the invoking account's rights profiles, actions can inherit privileges when they are launched from a trusted process. Therefore, the only way that actions can do privileged operation is if they have been assigned privileges in the account's profiles.
Actions are not managed by the Solaris Management Console.	Actions are assigned to rights profiles by the Rights tool. If new actions are added, the Security Administrator role needs to make the new actions available. See "To Make New Actions Available to the Rights Tool" on page 249.

Finding Which Privileges a Program Needs

The `runpd(1M)` command is the privilege debugging program for Trusted Solaris software. By default, the `runpd` command is in a rights profile assigned to the Security Administrator role. The environment for running the command is disabled by default.

The `runpd` command requires the trusted path attribute, which a command can only obtain when the command is executed by a role. Therefore, a user cannot use the `runpd` command to find out what privileges are required.

The root role should not be used to debug privilege use because the UID 0 may give the command more access than it would have with another UID. Similarly, the Primary Administrator role is not a good privilege tester. Executing `runpd` in any administrative role except root and `primaryadmin` logs the privileges needed by any normal user if the command is run at a label within the user accreditation range.

The procedure “To Find Out Which Privileges a Program Needs” on page 252 describes enabling the `runpd` environment and optionally creating an administrative role exclusively for doing privilege debugging.

The assignment of privileges should not be automatic. A program that fails due to lack of privilege may be assigned privileges, or, the Security Administrator role may decide to assign an effective UID or GID to make the privilege unnecessary.

When software has been assigned privileges or an alternate UID or GID, the software becomes *trusted* by virtue of the fact that it can bypass aspects of the Trusted Solaris security policy. Be aware that you can make software trusted even though it might not be worthy of trust. The Security Administrator role should not give any privileges to software until convinced that the software can use the privileges in a trustworthy manner. Only when it has been scrutinized and found to be using its privileges within the system security policy, can a program be called a trustworthy program.

Making Libraries Trusted

When a privileged program cannot find the libraries it requires, it fails with an error like the following:

```
ld.so.1: fatal: application-name: open failed: No such file or directory.  
Killed.
```

The Security Administrator role can add a privileged program's shared library directories to the list of trusted directories in `/var/ld/ld.config`. The `crle(1)` command must be used with both the `-u` and `-s` options followed by a colon-separated list of pathnames to the library directories.

- The `-u` option adds the library directories specified with the `-s` option to any previously-specified trusted directories.
- Entering `crle` without options displays the current trusted directories.
- Any other use of `crle` without the `-u` option removes previously-specified entries.

The `ldd(1)` command lists the libraries that are used by a program. See "To Make a Library Directory Trusted" on page 255 for how the Security Administrator can check for the library directories used by the application and run `crle` to add them to the trusted directories list.

The addition of a library directory to the list of trusted directories persists across reboots. However, if the `crle` command is ever entered with other options but without the `-u` option (perhaps by a third party script), any entries made on the command line will be removed.

To help ensure that all library directories needed for your privileged applications are configured as trusted directories at every reboot, the Security Administrator can create a boot-time script. See "To Make a Library Directory Trusted" on page 255 for how to create such a script.

See `/etc/rc2.d/S90wbem` for an example of adding the JAVA library directories needed by the SMC to the trusted library directories list. See `/etc/init.d/README` and `/etc/rc2.d/README` for naming and numbering conventions for boot scripts.

Note – By default, boot scripts run with the system shell and with a real UID of 0 at `ADMIN_LOW`.

Also see the `ld(1)` man page for information on the link editor for object files.

Adding Boot Commands

New commands can be added to run during boot in the following ways:

- In the `/etc/inittab` file.
See the `inittab(4)` man page and "Adding Commands to the `inittab` File" on page 243.

- In scripts in the `/etc/init.d` (linked to `/etc/rcn.d`) directory.
See the `init.d(4)` man page, and “Adding Commands to `/etc/init.d` Scripts” on page 243. Solaris behavior is described in “Run Control Scripts” in the *System Administration Guide, Volume I*.
- In the `/etc/inet/inetd.conf` file.
See the `inetd.conf(4)` man page and “Adding Services to the inet Daemon” on page 244.

Unless other attributes are explicitly configured, commands run from the `inittab` or from scripts in `/etc/init.d` during the boot process have a label and clearance of `ADMIN_LOW`, a real and effective UID and GID of 0, and no privileges. If commands added into one of the files in the previous list need non-default security attributes, the commands must be added to local rights profiles. The Rights tool should be launched from a toolbox with the Files scope on the computer whose local files have been modified. Profiles defined on a naming service master are not going to be available to the boot programs on a naming service client.

Adding Commands to the `inittab` File

Make the changes to the `inittab(4)` file, and then add the commands to the boot profile. For example, if a script in `/usr/local/bin` named `mysite` needs to run with real UID of root, the Security Administrator role does the following:

1. Uses the Admin Editor to add an entry for the `mysite` script to `/etc/inittab`:

```
lo:234:respawn:/usr/local/bin/mysite
```
2. Uses the Rights tool with the Files scope to add the script `/usr/local/bin/mysite` to the boot rights profile with a real UID of 0.

See “To Add Commands to the `/etc/inittab` File” on page 256.

Adding Commands to `/etc/init.d` Scripts

In the default Trusted Solaris environment, the `/etc/init.d` scripts are modified to use the system shell, `sysh(1M)`, instead of the Bourne shell, `sh(1)` when the service being started requires explicit privileges or other non-default security attributes that are defined in the boot profile. In the default boot scripts, `/bin/sysh` is used without the name of a profile argument because if no profile is specified, the system shell looks at the boot profile by default.



Caution – Do not modify the commands already specified in the boot profile or modify the default `/etc/init.d` scripts. You can either add new scripts or change only scripts that may be added when a new application imported to the system.

When additional commands need to run during boot with non-default security attributes, the Security Administrator role specifies the commands with the needed attributes either by creating a new boot-time rights profile or by modifying the existing boot profile using the SMC Rights tool.

The role also needs do one of the following in `/etc/init.d`: modify an existing shell script, or create a new shell script so that the script starts with `#!/sbin/sysh` as the first line.

See the README in the `/etc/init.d` directory and in each `/etc/rcn.d` directory for guidelines about the numbering of the scripts that start system services.

As shown in the following example, a system shell boot script has `#!/sbin/sysh` as the first line. If the Security Administrator role has added the needed commands into the boot profile, there is no need to specify a profile name. If the Security Administrator role has created a new boot profile, the second line has the `setprof` argument followed by the name of the *local_boot_profile*.

```
#!/sbin/sysh
setprof local_boot_profile
```

For example, if a command needs a process label other than `ADMIN_LOW`, the profile needs to specify the label and if the command needs a UID of root, the profile needs to specify the required UID. See “To Run `rc` Scripts With Security Attributes” on page 257.

Stopping or starting boot scripts in a Trusted Solaris environment requires privileges, so the script must be executed by the System Administrator Role in an administrative role workspace with the trusted path attribute, and the script’s name must be in one of the account’s rights profiles.

The toolbox from which the Rights tool is invoked should be running with the local Files scope on the computer where the script is added to the `/etc/init.d`.

Adding Services to the `inet` Daemon

Services started by `inetd(1M)` run with the label and clearance of the client. `inetd` also runs with other attributes of the client if the following are specified in `inetd.conf(4)`:

- If the *uid* field has the keyword `CLIENT`, the services start with the client’s UID, GID, primary and any secondary groups.

- If the *wait-status* field contains the `setaudit` flag, the services are started with the client's audit characteristics.

In addition:

- If the *wait-status* field contains the `trusted` flag, the trusted path attribute is available to the service.
- The Security Administrator can specify privileges and a label range by adding the service to the `inetd` rights profile and assigning the service the desired privileges and a label range.

If an entry in the `inetd` profile assigns privileges to the service, the service inherits the specified privileges.

If an entry in the `inetd` profile specifies minimum and maximum labels, `inetd` verifies that the label of the client is within the specified label range. If the label of the client is not the label range, the service is not executed.

See "To Add Services to the `inetd.conf` File" on page 258.

Managing Software (Tasks)

▼ To Mount a CD-ROM for Adding a Package

1. Assume the System Administrator role and go to an `ADMIN_LOW` workspace.
2. Allocate the CD-ROM device.
Use the Allocate Device option from the Trusted Path menu or launch the Device Allocation Manager action from the Tools subpanel in the Front Panel.
3. Double-click the name of the CD-ROM device in the list of Available Devices to transfer it to the list of Allocated Devices. Click OK to the `ADMIN_LOW` label.
4. Insert the CD-ROM at the specified label into the drive and click the OK button.
5. Click Yes to mount the CD.
6. Press return when prompted to close the window.
7. Deallocate the device when finished.

▼ To Give Forced Privileges to a Command

An executable file with forced privileges runs with those privileges when invoked in a profile shell by any user or role.

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**

A file's owner or a user with the Act as File Owner authorization can also add privileges to an executable at a label within the user or role's accreditation range.

2. **Navigate to the file's directory, and make sure the file is executable.**

If the file is not an executable and it should be, change permissions to make it executable, as in:

```
$ chmod 755 filename
```

3. **Give the command allowed privileges equal to the forced privileges you plan to assign.**

If you are using the File Manager Permissions dialog box, click the Allowed button, assign Allowed Privileges, and then click the Forced button to assign the Forced Privileges.

The following example shows using the `setfpriv(1)` command to set `file_dac_read` and `file_dac_write` as allowed and forced privileges.

```
$ setfpriv -s -f file_dac_read,file_dac_write \  
-a file_dac_read,file_dac_write test.priv.file
```

▼ To Create a New File Edit Action

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**

2. **Launch the Admin Editor action to open the `/usr/dt/appconfig/types/C/TSOLadmin.dt` file for editing.**

See "To Log In and Assume a Role" on page 25, and "To Edit a Local File" on page 33, if needed.

3. **Copy and paste the definition for one of the existing actions in the `TSOLadmin.dt` file.**

The example in this procedure modifies a copy of the `Vfstab` action.

```
ACTION Vfstab  
{  
    LABEL      Set Mount Points  
    ICON       Dtpenpd  
    TYPE       COMMAND  
    WINDOW_TYPE NO_STDIO  
    EXEC_STRING /usr/dt/bin/trusted_edit /etc/vfstab  
    DESCRIPTION Specify the file system mount points
```

```
}
```

4. Modify the copied action's definitions.

a. Change the ACTION name.

This example creates a new action to edit the `/etc/system` file to modify Trusted Solaris kernel switch settings.

```
ACTION EditSystemFile
{
```

b. Change the LABEL.

```
    LABEL          Edit System File
```

c. Change the ICON, if you have created a new icon or want to use another existing one from `/usr/dt/appconfig/icons/C`.

```
    ICON           Dtpenpd
```

d. Change the file name in the EXEC_STRING.

```
    EXEC_STRING    /usr/dt/bin/trusted_edit /etc/system
```

e. Change the text in the DESCRIPTION.

```
    DESCRIPTION    Modify system file
}
```

5. Save and close the `TSOLadmin.dt` file.

```
:wq
```

6. Copy and rename the `Vfstab` action file.

a. Go to `/usr/dt/appconfig/appmanager/C/System_Admin`.

b. Clone the `Vfstab` file and rename it to the name of the new action.

For example, rename `Vfstab` to `EditSystemFile`.

c. Make the action file executable.

Select the Permissions option on the File Manager's File menu and set the permissions to executable for owner, group, and other, or enter the following on the command line:

```
$ chmod 777 EditSystemFile
```

7. **In the System Administrator role, copy the modified `TSOLadmin.dt` and action files to every host in the domain.**

Since actions are not administered through the name service, some other means of distribution must be used, such as `rdist(1)` or `sneakernet` (copying the files to a floppy and carrying it around to install the files on each host).

8. **In the Security Administrator role, bring up the Solaris Management Console, choosing the appropriate name service scope.**

To make the action available only to one host, choose the Files scope on the host.

9. **Click Users and provide a password when prompted.**

10. **Double-click the Rights tool, then double-click either the Information Security profile or the Object Access Management profile.**

The Properties dialog box for the Right displays.

- a. **If the action edits a security-relevant file, such as the `/etc/system` file, open the Information Security profile.**

- b. **If the action edits an administrative file that would normally be modified by a UNIX system administrator and that does not contain labels or other security attributes, such as the `group` file, open the System Management profile.**

11. **Click Actions, then `System_Admin` in the Actions Denied column.**

The new action should be listed. Refer to the online help for assistance.

12. **Add the action to the rights profile, and assign to the action the same privileges that are assigned to the Set Mount Points action: `file_dac_read`, `file_dac_write`, `proc_audit_appl`, `proc_audit_tcb`.**

13. **To make the action usable, log out and log in again.**

▼ To Add Actions Outside of the `System_Admin` Folder

Adding actions can be done as described in the *Solaris Common Desktop Environment: Advanced User's and System Administrator's Guide*, within Trusted Solaris MAC constraints.

1. **As a user or role, create the action by using the `CreateAction` action, or manually.**
2. **Assume the Security Administrator role, and label the action `ADMIN_LOW`.**

3. Place it in the `/etc/dt/appconfig/types/C` directory so that CDE makes it available.

Note – Added actions are software. Their trustworthiness must be checked before they are made generally available. See “Testing New Software for Security” on page 235.

▼ To Make New Actions Available to the Rights Tool

1. Assume the Security Administrator role, and go to an `ADMIN_LOW` workspace.
2. Remove the `/usr/dt/appconfig/smc/C/actions` file on the SMC server.

```
$ rm /usr/dt/appconfig/smc/C/actions
```

3. Log out and log into the SMC server using the CDE login screen.
CDE regenerates the removed file, and populates it with all actions that it finds.

▼ To Write a Profile Shell Script

Note – When adding a profile shell script that runs commands with inherited privilege, the Security Administrator role needs to update an appropriate rights profile with a list of each of the commands that run within the shell script and to assign the commands any privileges they need. If a new shell script needs to be used by a role, all the commands that need security attributes must be added to the Custom *role_name* Profile or other profile that applies to the role, along with the name of the script itself.

Anyone with a text editor can write the shell script.

1. Start the script with `/bin/pfsh` (or any other profile shell) on the first line.

```
#!/bin/pfsh
```

2. Determine which commands need privileges and which privileges are needed.

In the example, `/usr/lib/fs/nfs/nfsfind` is a cron job owned by root that needs privileges in order to run successfully at `ADMIN_HIGH`. The `tfind` command needs the `file_dac_search` and `file_dac_read` privileges and the `rm` command needs the `file_dac_search`, `file_dac_write`, `file_dac_read`, and `file_mac_write` privileges. See “To Find Out Which Privileges a Program Needs” on page 252, if needed.

```
#!/bin/pfsh
# Copyright (c) 1993, 1997, 1998, 1999 by Sun Microsystems, Inc.
#ident  "@(#)nfsfind.sh 1.5      97/05/21 SMI; TSOL 2.x"
```

```
#
# Check shared NFS filesystems for .nfs* files that
# are more than a week old.
#
# These files are created by NFS clients when an open file
# is removed. To preserve some semblance of UNIX semantics
# the client renames the file to a unique name so that the
# file appears to have been removed from the directory, but
# is still usable by the process that has the file open.
if [ ! -s /etc/dfs/sharetab ]; then exit ; fi
  for dir in `awk '$3 == "nfs" {print $1}' /etc/dfs/sharetab`
do
  tfind $dir -M -name .nfs\* -mtime +7 -mount -exec rm -f {} \;
done
```

3. Assume the Security Administrator role and go to an ADMIN_LOW workspace.
4. Use the Rights tool to update an appropriate profile to list the script, each of the commands that need to run within the shell script and to assign the commands the privileges they need.

See “To Launch the Solaris Management Console” on page 30, if needed.

To continue with the example, to enable the System Administrator role to run the example cron script with the needed privileges, the Security Administrator uses the Rights tool to update the Custom Admin Role and makes sure it is assigned to the System Administrator role. The rights profile is modified to include the /usr/lib/fs/nfs/nfsfind script, the tfind command with the file_dac_search and file_dac_read privileges and the rm command with the file_dac_search, file_dac_write, file_dac_read, and file_mac_write privileges.



Caution – When you add commands to a profile and give them privileges or other security attributes, the commands execute with those attributes, not only in the profile shell script but whenever they are invoked in any profile shell, as long as the profile is in effect for the invoking account. The order of profiles is also important: the profile shell executes a command or action with whatever security attributes are specified in the first profile in the account’s list of profiles. For example, if `tfind` is in the Custom Root Profile with privileges, and the Custom Root Profile is the first profile in which `tfind` is found, then `tfind` will inherit the privileges specified in the Custom Root Profile when the root role executes `tfind` on the command line in a profile shell.

▼ To Write a Standard Shell Script that Runs Privileged Commands

Note – You can create a standard shell script to run its commands with privileges by adding the script to a rights profile and specifying the script to run with all the privileges that are needed by the script’s commands. The script then inherits privileges when invoked in a profile shell, when an account has a rights profile containing the script.

1. **Start the script with any standard shell (not a profile shell) on the first line.**

```
#!/bin/csh
```

2. **In the PrivDebug or the Security Administrator role, determine what privileges are needed by what commands in script.**

See “To Find Out Which Privileges a Program Needs” on page 252, if needed. The example, called `autosetpriv`, would enable the Security Administrator to assign a defined set of forced and allowed privileges to a file called `executable`. The `setfpriv` command in this script needs the `file_setpriv` privilege.

Note – This shell script is just an example. A normal shell script accepts the privileges and the filename as arguments and does error checking.

```
#!/bin/csh
setfpriv -s -f ipc_mac_write,proc_setsl,sys_trans_label
-a ipc_mac_write,proc_setsl,sys_trans_label executable
```

3. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**

4. Use the Rights tool to update an appropriate profile to list the script, each of the commands that need to run within the shell script and to assign to the commands the required privileges.

See “Adding or Modifying a Rights Profile” on page 87, if needed.

To enable the script called `autosetpriv` to run with the `file_setpriv` privilege needed by the `setfpriv` command, the Security Administrator role would use the Rights tool to update the Custom Secadmin Role (which is assigned to the Security Administrator role by default) to include the `autosetpriv` script and assign to `autosetpriv` the `file_setpriv` privileges.

5. Test, debug, and execute the shell script as desired in the profile shell.

```
$ autosetpriv
```

▼ To Save and Restore Privileges When Editing a File

1. Assume the Security Administrator role and use the `getfpriv` command to list the privileges on the executable file and save the output.

The following example directs the output into a temporary file.

```
$ getfpriv executable_file > tempfile
```

2. After editing the executable file, use the File Manager to make the file executable again (if needed) and then restore the privileges listed in the temporary file.

The following example uses the `setfpriv(1)` command to set the privileges stored in the `tempfile` created in step 1.

```
$ setfpriv -s -f `cat tempfile` program_pathname
```

▼ To Find Out Which Privileges a Program Needs

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.

2. (OPTIONAL) Create a Privilege Debug profile and assign it to an administrative role.

Note – The Security Administrator role is the only role assigned the `runpd` command in the default configuration. The administrative role that can be created in this step could be used exclusively for privilege debugging. The role can be used to find out what privileges a command needs when run by a normal user at a label within the user accreditation range or by an administrative role at one of the administrative labels.

a. In the appropriate scope, launch the SMC and use the Rights tool to create a new rights profile (such as Privilege Debug).

Give the new profile the commands: `/usr/sbin/runpd`, `/bin/getfpriv`, and `/bin/setfpriv`, the authorizations: Shutdown the System, Enable Login, and Set File Privilege, and the action: Admin Editor. The profile enables an administrative role to enable privilege debugging and execute the `/usr/sbin/runpd` command.

b. If desired, use the Administrative Roles tool to create an administrative role (such as PrivDebug).

c. Assign the role the new profile and the All profile, if desired.

Place the more restrictive rights profile, Privilege Debug, before the less restrictive profile, All.

d. If you created a new role, open the User Accounts tool and assign the privilege PrivDebug role to an account.

3. Assume the new role, or stay in the Security Administrator role.

4. Use the Admin Editor action to change the `tsol_privs_debug` setting to 1 in the `/etc/system` file.

```
set tsol_privs_debug=1
```

5. Use the Admin Editor action to remove the comment (#) at the beginning of the line that begins `kern.debug` in the `/etc/syslog.conf` file.

The following line logs the privileges requested by system calls and daemons in the `/var/log/privdebug.log` file.

```
kern.debug;daemon.debug;local0.debug /var/log/privdebug.log
```

6. Reboot to enable privilege debugging.

7. Log in and assume the Security Administrator role or, if you created a new privilege debugging role in step 2, assume the new role.

8. Create a workspace at the label at which the test program will typically be run.

9. In a terminal at the appropriate label, enter the `runpd` command followed by the name of the command and any options whose use of privilege you want to check.

As shown in the following example, `runpd` displays the name of the privilege(s) that the program needs in order to succeed followed by the type of access attempted (for example, `create`) followed by the name of the resource (for example, `RAW_SOCKET`).

```
$ runpd pathname_of_command_and_any_options

runpd: child terminated with a status of 0

process pathname_of_command pid process_ID
lacking privilege privilege_name
to perform type_of_access
upon resource resource_name (MM DD HH:MM)
```

The following example shows the result of running `runpd(1M)` on `ping(1M)` (for the purpose of the example, Trusted Solaris privileges were removed from the command).

```
$ runpd /usr/sbin/ping sif
sif is alive runpd: child terminated with a status of 0
process /usr/sbin/ping pid 5138 lacking privilege net_rawaccess
to create raw socket (Oct 25 18:33)
process /usr/sbin/ping pid 5138 lacking privilege
sys_net_config to manage transport opts (Oct 25 18:33)
```

10. Open an `ADMIN_HIGH` workspace, and check the log file for the privilege debugging messages.

A typical privilege debugging log entry is similar to the following:

```
$ cat /var/log/privdebug.log
Mar 29 12:18:43 hostname unix:
DEBUG: pathname_of_command pid
process_ID lacking privilege number to
number_of_type_of_access number_resource
```

The following screen shows the `privdebug.log` entries from when `runpd` was run on `ping`.

```
Oct 25 18:33:35 tern unix:
DEBUG: /usr/sbin/ping pid 5138 lacking privilege
36 to create raw socket
Oct 25 18:33:35 tribble unix:
DEBUG: /usr/sbin/ping pid 5138 lacking privilege
68 to manage transport opts
```

The privilege numbers appear after the word “privilege.” You can look up a privilege number in the `/usr/include/sys/tsol/priv_names.h` file to find its name. For example, the privilege number 36 is associated with the name `net_rawaccess`. The numbers following the privilege number and the word “to” are the number of the type of access attempted followed by the number of the resource.

11. Have the security administrator evaluate whether to give the program the privileges it is missing.

12. If the privileges should be granted, see “To Give Forced Privileges to a Command” on page 246 and “Adding or Modifying a Rights Profile” on page 87 for how to assign privileges.
13. Turn off privilege debugging: restore the changes you made to the `/etc/system` file and the `/etc/syslog.conf` file in step 3 and step 5 and reboot the machine.

▼ To Make a Library Directory Trusted

The following procedure describes how to determine which libraries should be trusted, and then describes how to add them to the trusted library directories list.

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.

2. Remove any forced privileges assigned to the program.

If the program does not have forced privileges, then you do not need to do the following substeps or step 4.

- a. Check the command for forced privileges.

The following example gets the list of privileges and saves them in a file. The file can be used to reset the privileges, if any, after the `ldd(1)` command is run.

```
$ getfpriv -s -f program_name > filename
```

- b. Remove the privileges from the program.

```
$ setfpriv -s -f none program_name
```

3. Use the `ldd` command to find out which library directories the application program is using.

```
$ ldd program_name
```

Note – The program that is using the shared libraries may be imbedded in a script, so make sure you are running `ldd` on the actual program that needs the libraries.

4. If you removed forced privileges from the program in step 2, add the privileges back to the program.

The following example uses `setfpriv` to set the privileges stored in `filename` in step a.

```
$ setfpriv -s -f `cat filename` program_name
```

5. Use the Rights tool to add the `crle` command to the Custom Secadmin Profile with a real UID of 0, and a label and clearance of `ADMIN_LOW`.

See the `crle(1)` man page for more information.

6. Use the `crle` command to add the library directories from step 3 to the list of trusted library directories.

```
$ crle -u -s [directory_1[: . . . :directory_N]
```

7. To regenerate the list of trusted directories at every reboot, add the `crle` command to a boot-time script.

Use the Admin Editor to create or modify a script in the `/etc/init.d` directory.

If a privileged application already has a script, modify the existing script. Otherwise, create a new script.



Caution – Do not modify any default Trusted Solaris scripts. Modify only scripts that are installed with new applications that need privileges.

The following is an example of executing the `crle` command in a script:

```
crle -u -s directory_1[: . . . :directory_N]
```

8. Make a hard link from the script in the `/etc/init.d` directory.

Use the `S` prefix in the target file's name for starting the script. Use the proper two-digit number in the target file's name to determine the order in which the script is executed during the run level. See the README in `/etc/init.d` and `/etc/rc2.d`, if needed.

In the following example, the name of the new script in `/etc/init.d` is `new_script`, which is linked to `/etc/rc2.d/S87new_script`.

```
$ cd /etc/rc2.d
$ ln /etc/init.d/new_script S87new_script
```

▼ To Add Commands to the `/etc/inittab` File

1. Assume the Security Administrator role and use the Admin Editor action at **ADMIN_LOW** to edit the file `/etc/inittab`.

The following example adds `/usr/local/bin/mysite` to the file.

```
lo:234:respawn:/usr/local/bin/mysite
```

2. Save and quit the file.

```
:wq
```

3. Use the Rights tool in the Files scope to add the script `/usr/local/bin/mysite` to the `boot` rights profile with a real UID of 0.

▼ To Run rc Scripts With Security Attributes

Note – To add security to rc scripts, create a new profile and use the `setprof` command to refer to the new profile in a new `sysh(1M)` script, as described in this procedure.

1. Assume the Security Administrator role and use the Admin Editor action at **ADMIN_LOW** to create a new `sysh` script in `/etc/init.d`.

See the `sysh(1M)` man page and “System Shell” on page 227. The first line of the script should read as follows:

```
#!/bin/sysh
```

2. On the second line of the script, type in the `setprof` option to identify the name of a rights profile.

```
setprof new_profile_name
```

3. Save and quit the file.

```
:wq
```

4. In the `/etc/init.d` directory, make a hard link from the new script to the desired `/etc/rcn.d` directories.

- a. For each run level at which the command should be started or stopped, go to the appropriate `/etc/rcn.d` directory and create a hard link from a properly-named target file to the `/etc/init.d` directory.
- b. Use the proper prefix in the target file’s name for either starting (S) or stopping (K).
- c. Use the proper numbers in the target file’s name to help determine the order in which the script is executed during the run level.

In the following example, the name of the new script in `/etc/init.d` is `new_script`, which is linked to `/etc/rc2.d/S89new_script` and `/etc/rc2.d/K89new_script`.

```
$ pwd
/etc/init.d
$ ln new_script /etc/rc2.d/S89new_script
$ ln new_script /etc/rc2.d/K89new_script
```

5. Use the Rights tool with the Files scope to create a new rights profile.
6. Add the command and any desired security attributes.
7. Reboot to effect the change.

▼ To Add Services to the `inetd.conf` File

1. Assume the Security Administrator role and use the Admin Editor action in an **ADMIN_LOW** workspace to open the `/etc/inet/inetd.conf` file for editing.

For example, the following line adds a service in `/usr/local/bin` named `newservice` with the `CLIENT` keyword in the `UID` field so that the service executes with the `UID` and `GID(s)` of the `CLIENT`. In the `flags` field, the `trusted` keyword causes the service to run with the trusted path attribute, and the `setaudit` keyword causes the service to run with the client's audit characteristics:

```
myport stream tcp6 nowait,trusted,setaudit CLIENT /usr/local/bin/newservice
```

2. Save and quit the file.

```
:wq
```

3. If the service needs to run with privileges or a restricted label range, use the Rights tool to add the service to the `inetd` rights profile along with any desired security attributes.

▼ To Install a Java Jar File

The security administrator must verify that the source of the Java program is trustworthy, that the method of delivery is secure, and that the program can run in a trustworthy manner. To limit the security risk, the system administrator can download the software to a single label within the users' accreditation range, where the security administrator can test it at that label. The administrator can then downgrade the label to `ADMIN_LOW`, and install it on an application server to make it available to all users.

1. Assume the System Administrator role and go to an **ADMIN_LOW** workspace.

2. Download the Java jar file to the `/tmp` directory.

For example, if you are selecting software from the `http://www.sunfreeware.com`, download the link that is described with "Web Start Wizard form of *application*."

3. Open a File Manager to the `/tmp` directory.

The Software Installation profile includes the Open action for Java code.

4. Double-click the downloaded file.

5. Answer the questions in the dialog boxes to install the software.

6. Read the installation log.

Index

A

access

- administrator responsibilities, 41
- to devices, 203
- to printers, 189

access policy

- devices, 204

accounts

- assigning labels, 87
- assigning passwords, 86
- assigning rights, 86
- assigning roles, 87
- assigning shells, 91
- deletion precautions, 43
- planning, 63
- security precautions, 43
- startup files, 67, 79

accreditation checks, 136, 139

actions

- adding new, 246
- adding outside the System_Admin folder, 248
- restricted by account profiles, 228
- using, 240

Add to NIS+ Administrative Group

- action, 102, 110, 185

add_allocatable command, 215

Admin Editor

- using, 33

ADMIN_LOW label

- installing software, 225
- mail, 106

ADMIN_LOW label (*continued*)

- protecting administrative files, 42
- role workspace, 23

administrative actions

- Add to NIS+ Administrative Group, 102, 110, 185
- adding, 248
- Admin Editor, 33
- available in Rights tool, 249
- creating, 246, 248
- in System_Admin folder, 32
- launching remotely, 36
- Name Service Switch, 147
- name services, 188
- Set Mail Options, 111
- trusted, 228
- using, 32

administrative roles

- adding a name service client, 185
- administering NIS+, 102
- administering remotely, 36
- assuming, 23, 25
- changing workspace label, 34
- described, 95
- exiting, 29
- launching the Printer Administrator, 196
- logging in remotely, 35
- remote role assumption, 103
- saving and restoring name service databases, 186
- workspaces, 23

- Administrative Roles tool
 - using, 102
- administrative tools, See Solaris Management Console, 30
- adminvi command
 - aliasing vi, 98
- allocate command
 - described, 215
- allocate error state
 - caused by failure of eject, 214
 - defined, 211
 - procedure for correcting, 218
- Application Manager
 - as trusted process, 228
- applications
 - assigning forced privileges, 246
 - evaluating for security, 236
- at command
 - administrative differences, 73
- at jobs
 - running privileged commands, 74
- at.allow file, 75
- atohexlabel command, 61
- atq command, 73
- atrm command, 73
- attr_mac_policy, 204
- audio coprocessor, 213
- AUDIO_DRAIN ioctl
 - run by device_clean, 213
- AUDIOGETREG ioctl
 - run by device_clean, 213
- AUDIO_SETINFO ioctl
 - resetting device to default, 213
- auth_name file, 49
- authorizations
 - adding, 48
 - adding to software, 57
 - Allocate Device, 203, 210, 222
 - device-related, 210
 - device-related procedures, 222
 - Edit Owned Jobs, 76
 - for device administration, 206
 - for devices, 220
 - Manage All Jobs, 76

B

- banner pages
 - printing without, 200
- batch command, 73
- boot rights profile, 243

C

- cachefs filesystem type, 168
- CD-ROM
 - mounting, 245
- CD-ROM devices
 - accessing, 203
 - device_clean script, 212
 - launching audio program, 180
 - mounting, 180
- CIPSO
 - use in packets, 132
- commands
 - giving forced privileges, 246
 - privileges, 241
 - trusted, 241
- commercial applications
 - evaluating, 236
- .copy_files file
 - example, 78
 - setting up for users, 77
 - using, 72
- cron command, 73
- cron jobs
 - running privileged commands, 74
- cron.allow file, 75
- cron.deny file, 75
- crontab command, 73
- cut and paste
 - configuring selection transfer rules, 45

D

- DAC
 - device files, 215
 - policy for devices, 204
- deallocate command, 215
- default shells
 - assigning to accounts, 91

- developers responsibilities, 239
- Device Allocation Manager
 - administering devices, 207
 - allocating and administering devices, 223
 - allocating devices, 206
- device policy
 - setting, 217
- device special files
 - access policy, 204
- device_allocate file, 215
- device_clean command, 215
- device_clean scripts
 - for tape devices, 212
 - modifying, 224
 - procedure for adding devices, 218
 - review, 211
- device_maps file, 215
- device_policy file, 205
- devices
 - access policy, 204, 205
 - accessing, 206
 - adding, 218
 - adding device_clean script, 224
 - adding site-specific authorizations, 220
 - administering, 204, 223
 - authorizations, 210
 - configuring serial line, 220
 - modifying policy, 217
 - non-allocatable
 - setting the label range, 204
 - policy defaults, 204
 - reclaiming, 218
 - setting policy, 204
 - setting up audio, 218
- /dev/kmem kernel image file
 - security violation, 238
- directories
 - changing flags, 163
 - changing labels and privileges, 175
 - security attributes, 161, 166
 - sharing, 179
 - upgraded, 160
- dminfo command
 - reporting entry in the device_maps, 215
- dtsession command
 - running updatehome, 72

- dtterm terminal
 - forcing the sourcing of .profile, 80, 98
- dtwm command, 228

E

- Edit Owned Jobs authorization, 76
- editing privileged executables, 235
- email
 - managing, 105, 120
 - options, 106
 - switching mail tools, 107, 120
 - troubleshooting, 108
- emetric
 - described, 141
- emetrics
 - using in routing, 155
- /etc/cron.d/CRON
 - cron lock file, 75
- /etc/cron.d/cron.admin file
 - creating, 81
- /etc/default/login file
 - specifying RETRIES, 44
- /etc/init.d directory
 - RMTMPFILES script, 55
- /etc/init.d scripts
 - Trusted Solaris modifications, 243
- /etc/nologin file
 - disabling logins, 55
- /etc/skel directory, 71
- exec system call
 - inheriting privileges across, 230
- executable files
 - assigning forced privileges, 246
 - editing while preserving privileges, 235
- exporting software, 225

F

- failsafe session
 - recovering from startup file errors, 70
- fallback mechanism
 - creating, 150
- FDFS
 - mounting, 167

- file_mac_write privilege
 - resulting in a file's dominating its directory's SL, 57

- File Manager
 - as trusted process, 228
 - changing security attributes, 161
 - Privileges dialog box, 229

- file systems
 - cachefs type, 168
 - changing security attributes using mount, 178
 - changing security attributes using newsecfs, 176
 - changing security attributes using setfsattr command, 177
 - changing security attributes using vfstab file, 178
 - fdfs type, 167
 - hsfs type, 167
 - lofs type, 167
 - managing, 160
 - nfs type, 167
 - pcfs type, 167
 - security attributes, 163, 166
 - sharing, 179
 - single label, 165
 - table of supported types, examples, notes, 168
 - tmpfs type, 168

- file_upgrade_sl privilege
 - resulting in upgraded names, 57

- files
 - backing up, 174
 - changing flags, 163
 - changing labels, 162
 - changing privileges, 162
 - managing, 160
 - procedure for changing labels and privileges, 175
 - restoring, 174
 - upgraded, 160

- floppy disk devices
 - accessing, 203
 - device_clean script, 212

- forced privileges
 - assigning, 246

- fork system call
 - inheriting privileges across, 230

- Front Panel
 - as trusted process, 228
 - Device Allocation Manager, 206

G

- getdents system call
 - restricting from returning upgraded names, 57

- getfattrflag command, 163

- getfpriv command
 - using to save privileges, 252

- getfsattr command, 165

- groups
 - deletion precautions, 43
 - security requirements, 42

H

- help file
 - creating, 89

- hexadecimal label equivalents
 - determining, 61

- host types
 - networking, 125
 - table of templates and protocols, 125

- hosts
 - networking concepts, 121

- HSFS
 - mounting, 167
 - procedure for mounting, 180

I

- icons
 - visibility
 - in the File Manager, 228
 - in the Workspace Menu, 228

- identification and authentication
 - before assuming a role, 23, 24

- IMAP server
 - adding to NIS+ admin group, 110

- inheritable privileges, 230
- init.d directory
 - RMTMPFILES script, 55
- initialization files
 - Trusted Solaris differences, 68
- Interface Manager tool
 - using, 154
- internationalization
 - changing printer output, 195
- IP Options
 - using for trusted routing, 129

J

- Java jar files
 - installing, 258

K

- kadb command
 - overriding default setting, 54
- kernel switches
 - changing defaults, 56
 - configurable, 56
- keyboard shutdown
 - changing default, 54
 - enabling, 54
- kmem kernel image file, 238

L

- label ranges
 - receiving mail outside of, 110
 - setting on individual computers, 204
 - setting on printers, 204
- label_encodings file
 - procedures
 - printing without banners and trailers, 200
- labels
 - changing on files and directories, 175
 - seeing on directories, 61
- libt6 library, 75

- .link_files file
 - using, 72
 - example, 79
 - setting up for users, 77
- links - symbolic
 - MAC attributes, 160
- list_devices command, 215
- local.login file
 - defining printers, 202
- LOFS
 - mounting, 167
- login
 - by administrative roles, 23, 35
 - configuring serial line, 220
 - maximum allowed number of failures, 44
 - opening an account closed by too many failed logins, 44
 - preventing being disabled after reboot, 55
 - setting the maximum number of failures, 44
- .login file
 - setting up for users, 79, 80
- login sequence, 55
- login shells
 - assigning to roles, 97

M

- MAC
 - cautions about override privileges, 237
 - incoming packets
 - packets, 139
 - outgoing packets, 137
 - policy for devices, 204
- mail
 - adding IMAP server, 110
 - alternate application, 116
 - checking network connections, 113
 - creating action, 116
 - installing alternate mailer, 119
 - loss of mail icons, 116
 - managing, 105, 120
 - modifying an alias, 111
 - options, 106, 111
 - outside label range, 110
 - setting up IMAP server, 110
 - substituting alternate application, 118

- mail (*continued*)
 - switching mail tools, 107, 120
 - troubleshooting, 108, 112
 - viewing the mail queue, 111
- Mailing Lists tool
 - using, 111
- .mailrc file, 70
- man pages
 - accessing all, 72
- Manage All Jobs authorization, 76
- MANPATH environment variable, 72
- MLDs
 - listing user's home directories, 61
 - mounting, 160
 - mounting on unlabeled hosts, 160
 - privilege requirements, 160
- mounts
 - managing, 160
 - procedure for TMPFS file systems, 180
 - troubleshooting, 181

N

- Name Service Switch action, 147
- name services
 - actions for managing, 188
 - adding a client, 185
 - advantages, 183
 - databases unique to Trusted Solaris
 - environment, 185
 - managing, 183
 - saving and restoring databases, 186
- network interfaces
 - configuring, 154
 - requirements, 135
- networking concepts, 121
- networks
 - default labeling, 138
 - using templates, 146
- newsecfs command, 176
- NFS
 - mounting, 167
- NIS+
 - adding a NIS+ client, 185
 - adding IMAP server, 110
 - adding role to admin group, 102

- NIS+ (*continued*)
 - saving and restoring tables, 186
 - viewing table information, 188
- normal user
 - accessing devices, 203
- nsswitch.conf file
 - trusted network configuration, 147

O

- object reuse
 - clearing names of empty directories, 160
- open_priv
 - policy for devices, 204

P

- packages
 - accessing the CD, 245
- packets
 - IP options, 129
 - IP options field, 129
 - outgoing
 - MAC rules, 137
 - security attributes, 134
- passwords
 - assigning, 86
 - changing allowed tries, 53
 - role, 23
 - storage, 42
- PCFS
 - mounting, 167
- permissions
 - on devices, 204
- polling trusted network databases
 - changing, 150
- Printer Administrator
 - launching, 196
- printers
 - setting label range, 204
- printing
 - accessing remote printer, 200
 - configuring attached printer, 196
 - configuring for labels, 197
 - configuring labels and text, 195

- printing (*continued*)
 - managing, 189
 - restricting label range, 198
 - using a non-Trusted Solaris server, 196
 - without banners and trailers, 200
 - without labels, 195
 - without page labels, 201
- privilege debugging
 - setting tsol_privs_debug, 57
- privileged commands
 - run by cron and at, 74
- privileged programs, 241
- privileges
 - adding, 49
 - adding to software, 59
 - allowed, 230
 - assigning forced, 246
 - changing on files and directories, 175
 - debugging, 252
 - forced
 - assigning, 246
 - giving forced, 229
 - inheritable, 230
 - non-obvious reasons for requiring, 238
 - passing to child processes, 230
 - saving and restoring an edited
 - executable, 252
- priv_name file, 50
- priv_names.h file, 50
- PROCFS
 - mounting, 168
- .profile file
 - setting up for users, 79, 80
- profile shell
 - enabling privilege, 233
 - startup algorithm, 70
- profiles
 - assigning, 86
- programs
 - commercial
 - assigning privileges to, 229
 - new, trusted, 229
 - privilege debugging, 252
 - trusted vs. trustworthy, 241

R

- rcp command
 - required privilege, 237
- real UID of root
 - required for applications, 238
- reboot
 - changing device_policy, 205
- remote administration
 - editor limitations, 100
- remote logins
 - enabling for roles, 35
- remote role assumption, 103
- remove_allocatable command, 215
- rights
 - assigning, 86
- rights profiles
 - boot, 243
 - controlling the use of actions, 228
 - creating, 89
 - creating new for boot commands, 243
 - listing, 88
 - modifying, 90
- Rights tool
 - specifying privileges for commands and
 - actions, 230
 - using, 81, 89, 90, 101
 - viewing new actions, 249
- RIPSO
 - supported classifications, 133
 - use in packets, 133
- roles
 - administrative, 95
 - assigning login shell, 97
 - creating, 96, 102
 - listing, 100
 - managing, 95
 - modifying, 101
 - See administrative roles, 23
- root UID
 - required for applications, 238
- routers, 140
- routing
 - concepts, 139, 143
 - static with emetrics, 155
 - tables, 141
- run control scripts
 - shell use, 227, 243

runpd command, 57

S

/sbin/sysh shell, 243

security administrators

enforcing security, 211

modifying window configuration files, 52

security attributes

file systems, 161, 166

modifying user defaults, 77

saving to tape, 216

setting at mount time, 178

setting for remote hosts, 149

setting on file system, 177

setting using newsecfs, 176

Security Families tool

assigning templates, 149

using, 148

security features

identification and authentication, 23

security mechanisms

extendable, 48

security policy

allowing a wildcard in special boot

files, 147

training users, 40, 211

sel_config file

changing defaults, 57

configuring selection transfer rules, 45

sections, 47

sel_mgr command, 47

sendmail command

tracing mail delivery, 112

using, 108

serial line

configuring for logins, 220

Set Mail Options action, 111

setfattrflag command, 163

setfpriv command, 229

setfsattr command, 165

shell scripts

profile, 234

Trusted Solaris behavior, 233

user and role requirements, 234

writing, 249

shell scripts (*continued*)

writing privileged, 249

writing privileged using standard
shells, 251

shells

assigning to accounts, 91

assigning to roles, 97

profile, 228, 249

profile startup algorithm, 70

standard, 232

sysh, 243

skeleton directories

defining printers, 202

use in Trusted Solaris, 71

software

exporting, 225

importing, 225

installing at ADMIN_LOW, 225

installing Java programs, 258

porting

reasons against, 238

privilege debugging, 252

Solaris Management Console

Administrative Roles tool, 102

Interface Manager tool, 154

launching, 30

Rights tool, 81, 89, 90, 101

Security Families tool, 148

User Accounts tool, 92

startup files

configuring accounts, 67, 79

.mailrc file, 70

procedures for customizing, 79

read at window system startup, 68

recovering from errors, 70

RMTMPFILES, 55

Stop-A

changing default, 54

enabling, 54

str_type, 204

symbolic links

MAC attributes, 160

sysh shell, 243

System_Admin folder

using administrative actions, 228

system file

changing defaults, 56

- system security
 - violations, 238
- system shell
 - enabling privilege, 227

T

- tape devices
 - accessing, 203
 - device_clean scripts, 212
- tar
 - saving security attributes, 216
- TMPFS
 - mounting, 168
 - procedure for mounting, 180
- tnd polling interval
 - changing, 150
- Tools subpanel
 - Device Allocation Manager, 206
- troubleshooting
 - loss of mail icons, 116
 - mail delivery, 112
 - mounts, 181
 - sendmail, 108
- trusted networking
 - 0.0.0.0 tnrdhdb entry, 151
 - fallback mechanism, 150
 - host types, 125
- trusted path attribute
 - when available, 95
- Trusted Path menu, 23
- trusted processes
 - defined, 228
 - launching actions, 228
- trusted programs
 - adding, 238
 - defined, 241
- trusted_edit script
 - assigning as default editor, 99
- trustworthy programs, 241
- tsol_hide_upgraded_names kernel switch, 57
- tsol_privs_debug kernel switch, 57
- TSOLadmin.dt file
 - adding an administrative action, 246
- tsolgateways file, 155

- tunnel file
 - procedure for creating, 156
 - setting up tunneling, 148
- tunnelling
 - passing emetrics through non-TSOL hosts
 - gateways, 148

U

- UFS
 - mounting in Trusted Solaris, 168
- UIDs
 - effective UID of root, 238
- UNIX domain socket
 - used by cron and its clients, 75
- unlabeled hosts
 - mounting MLDs, 160
- updatehome command, 72
- upgraded names, 57
- User Accounts tool
 - assigning rights profiles, 230
 - opening an account closed by too many
 - failed logins, 44
 - using, 92
- User Templates tool
 - advantages, 84
 - using, 91
- users
 - access to devices, 203
 - access to printers, 189
 - assigning authorizations, 93
 - assigning rights, 93
 - creating, 84, 92
 - creating templates, 84, 91
 - modifying, 92
 - modifying security defaults, 77
 - preventing account locking, 53, 54
 - security training, 39, 43, 211
 - setting up skeleton directories, 80
 - setting up startup files, 77, 79
 - tracking others' jobs, 81
 - /usr/dt/appconfig/appmanager/C/System_Admin
 - file
 - adding an administrative action, 247
 - /usr/dt/appconfig/types/C/TSOLadmin.dt
 - file

users (*continued*)

- adding an administrative action, 246

V

vi command

- aliasing to trusted_edit, 100

W

window manager, 228

window system

- trusted processes, 228

Workspace Menu

- as trusted process, 228

- customizing, 60

X

Xtsolusersession script, 228