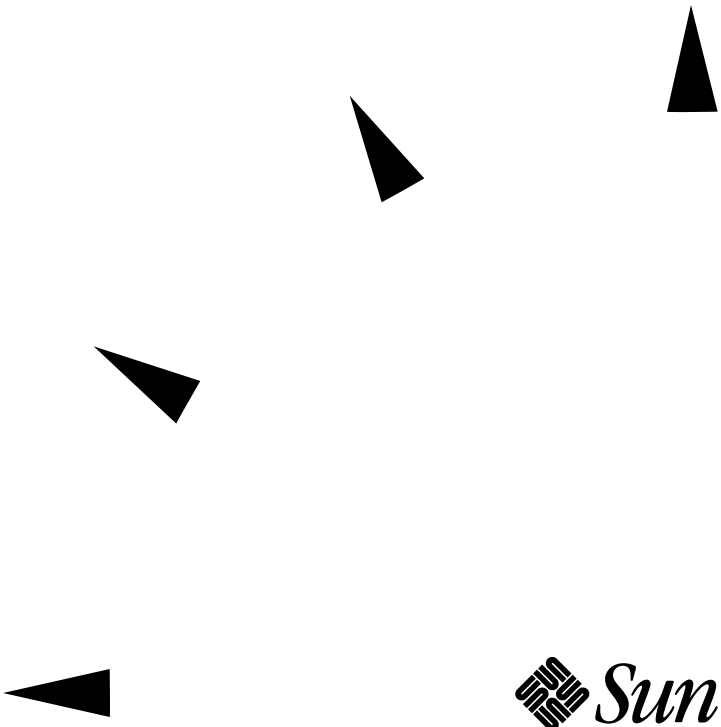
# Trusted Solaris Administrator's Procedures



**Sun Microsystems Federal, Inc.**
A Sun Microsystems, Inc. Business
901 San Antonio Road
Palo Alto, CA 94303
U.S.A.

Part No.: 805-8008-10
Revision A, July 1997

Sun microsystems

THE NETWORK IS THE COMPUTER™

Copyright 1997 Sun Microsystems, Inc. 2550 Garcia Avenue, Mountain View, California 94043-1100 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, SunSoft, SunDocs, SunExpress, and SunOS, OpenWindows, NFS, Sun Ultra, Ultra, JumpStart, Solaris, Solstice, Solstice AdminSuite, Solstice AdminTools, Solstice Autoclient, Solstice CacheOS, Disksuite, ToolTalk, X11/NeWS, Trusted NeWSprint, IPC, OpenBoot, SHIELD, XView, SunInstall, and Trusted Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. X/Open® is a registered trademark and "X" device is a trademark of X/Open Company Limited, Netscape is a trademark of Netscape Communications Corporation, and PostScript is a trademark of Adobe Systems, Incorporated.

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

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

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

---

Copyright 1997 Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, Californie 94043-1100 Etats-Unis. Tous droits réservés.

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

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

Sun, Sun Microsystems, le logo Sun, SunSoft, SunDocs, SunExpress, et Solaris SunOS, OpenWindows, NFS, Sun Ultra, Ultra, JumpStart, Solstice, Solstice AdminSuite, Solstice AdminTools, Solstice Autoclient, Solstice CacheOS, Disksuite, ToolTalk, X11/NeWS, Trusted NeWSprint, IPC, OpenBoot, SHIELD, XView, SunInstall, et Trusted Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. .X/Open® est une marque enregistrées et "X" device est une marque de X/Open Company Limited, Netscape est une marque de Netscape Communications Corporation, et PostScript est une marque de Adobe Systems, Incorporated.

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

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

Please
Recycle

Adobe PostScript

# Contents

Preface . . . . . xlvii

*Part 1 —Procedures Common to All Tasks and Administrative Roles*

**1. Assuming a Role and Working in a Role Workspace . . . . . 3**

Review of Administrative and Non-administrative Role Concepts 4

Administrative Roles . . . . . 4

Non-administrative Roles . . . . . 4

Logging In and Assuming a Role . . . . . 4

Auditing of Administrative Actions . . . . . 5

How Logins are Enabled . . . . . 5

Preventing Logins From Being Disabled After a Reboot . . 6

Working in the Administrative Role Workspace . . . . . 7

Application Manager Folders and Actions Icons . . . . . 8

Using Administrative Applications in Solstice_Apps Applications Folder . . . . . 8

Using Administrative Actions in the System_Admin Applications Folder . . . . . 8

Accessing the Application Manager . . . . . 9
Accessing Commands and Actions . . . . . 11
Using the Profile Shell To Do Other Tasks . . . . . 11
Administrative `vi` . . . . . 12
Administrative Role Procedures . . . . . 12
▼ To Log In and Assume an Administrative Role. . . . . 12
▼ To Switch Among Administrative Role Workspaces and the Normal User Workspaces . . . . . 21
▼ To Work at Multiple Labels While in an Administrative Role . . . . . 21
▼ To Launch Administrative Applications . . . . . 23
▼ To Launch Administrative Actions . . . . . 24
▼ To Create A New Administrative Action for Editing an Administrative File . . . . . 24
▼ To Add Actions Outside the System_Admin Folder. . . 26
▼ To Prevent Logins From Being Disabled After a Reboot 26
**2. Miscellaneous Tasks and Procedures . . . . . 29**
Getting a Hexadecimal Equivalent for Labels and Clearances 30
▼ To Get a Hexadecimal Equivalent for a CMW Label, an SL, IL, or Clearance . . . . . 30
Extending Extendable Security Mechanisms . . . . . 31
Understanding Authorizations . . . . . 32
Extending the Trusted Solaris Authorizations . . . . . 33
`auth_names.h` . . . . . 33
`auth_name(4TSOL)` . . . . . 35
▼ To Add An Authorization . . . . . 36

Extending the Trusted Solaris Privileges . . . . . 37
`priv_names.h` . . . . . 37
`priv_names(4TSOL)` . . . . . 39
▼ To Add a Privilege . . . . . 40
Changing the Trusted Stripe . . . . . 42
▼ To Change the Font or Color of the Trusted Path Indicator 42
Working with MLDs . . . . . 43
MLD Prefix / MLD Adornment . . . . . 44
How SLDs Are Created . . . . . 44
How SLDs Are Named . . . . . 44
Restriction on the Creation of MLDs and Its Effects . . . . . 45
MLD and SLD Prefixes . . . . . 46
Creating, Changing, Finding Your Way Around In, and Deleting MLDs . . . . . 46
▼ To Find Out if a Directory is an MLD . . . . . 49
▼ To Create an MLD from the File Manager . . . . . 49
▼ To Create an MLD from the Command Line . . . . . 49
▼ To Identify an MLD . . . . . 50
▼ To Identify an SLD . . . . . 51
▼ To Address the Entire MLD . . . . . 51
▼ To Change the MLD Prefix of a File System . . . . . 51
▼ To Remove an MLD . . . . . 52
*Part 2 —Administering Users, Roles, Profiles and Mail*
**3. Managing User Accounts . . . . . 57**

Things to Do Before Setting Up Accounts . . . . . 58

Decisions To Make Before Setting Up User Accounts . . . . . 58

How Responsibilities for Managing Users Are Divided. . . . . 60

Managing Users: Divided Between Two Administrative Roles 61

System Administrator Responsibilities. . . . . 61

Security Administrator Responsibilities . . . . . 61

Alternatives to Two-Role Administration. . . . . 62

Authorizations for Access to Account Management Tasks 62

Managing Startup Files in a Trusted Solaris System . . . . . 65

Controlling Which Startup Files Are Read While the Window System is Coming Up . . . . . 65

How the Reading of Start Up Files Is Controlled for the Profile Shell User . . . . . 67

Startup Files Launched When a Shell Comes Up, . . . . . 68

Skeleton Directories. . . . . 69

Using `.copy_files` and `.link_files`. . . . . 70

If `.copy_files` is Used to Copy Files Between SLDs: 70

If `.link_files` is Used to Link Files Between SLDs: 71

Worksheet for Copy and Link Files . . . . . 72

Administering the Automatic Running of Jobs Using `cron`, `at`, and `batch` . . . . . 73

Review of Concepts . . . . . 73

`crontab` Files. . . . . 73

`atjob` Files . . . . . 74

Supporting Jobs at Multiple Labels in the Spool Directories . . . . . 74
Determining Whether the Profile Shell is Used By a Job . . 74
Running Privileged Commands in `at` or cron Jobs . . . . . 75
Using a UNIX Domain Socket for Communications . . . . . 76
Ancillary Files . . . . . 76
Access to `at` and `cron` . . . . . 76
Allowing Access to Jobs Owned by Others . . . . . 77
`at.admin` and `cron.admin` Files . . . . . 78
Conditions for Access to Other's Jobs . . . . . 78
Conditions for `at`-related Commands . . . . . 78
Conditions for the `crontab` Command . . . . . 79
Changes to `crontab`(1TSOL) . . . . . 80
Changes to the `at` Command . . . . . 81
Changes to the `atq` Command . . . . . 81
Changes to the `atrm` Command . . . . . 82
Miscellaneous . . . . . 82
User Setup Procedures . . . . . 83
▼ To Make `.login` and `.profile` Looked at During Login for Bourne, C and Korn Shell Users . . . . . 83
▼ To Separate the Shell Initialization Files for Each Shell 83
▼ To Propagate Startup Files to Home Directory SLDs . . 84
**4. Managing Roles . . . . . 87**
Differences Between Role Accounts and User Accounts . . . . . 88

Differences Between Administrative and Non-Administrative Role Accounts . . . . . 88
Non-administrative Roles . . . . . 88
When to Create a Non-administrative Role . . . . . 89
Administrative Roles . . . . . 89
When to Create a New Administrative Role . . . . . 90
Things That Need the Trusted Path Attribute . . . . . 90
Dividing the Tasks of Managing User and Role Accounts . . . . 91
Authorizations for Access to Account Management Tasks . . . 92
Authorization for Specifying Information for One's Own Role 94
Alternatives to Two-Role Administration . . . . . 94
Creating a New Role . . . . . 95
Required Privileges . . . . . 95
Override Privileges . . . . . 96
DAC Override Privileges . . . . . 96
MAC Override Privileges . . . . . 96
Options for Avoiding the Need for Privilege . . . . . 96
Verifying the Use of Security Attributes Within Security Policy . . . . . 97
Example: Using the Man Page When Configuring `mount` in a Profile . . . . . 97
▼ To Configure a New Role . . . . . 98
**5. Using the User Manager to Configure User and Role Accounts 99**
Understanding the Information Entered in the User Manager Dialog Boxes . . . . . 100
Identity . . . . . 102

User Name, User ID, Group Name(s) and Group Id(s) 102
Comment . . . . . 103
Shell . . . . . 104
Account Type . . . . . 105
Password . . . . . 106
Background About Creating a Password or Selecting Other Password Options . . . . . 106
Background on the Password Duration and Warning Fields . . . . . 109
Background About Selecting a Method for Password Generation . . . . . 109
Background on the Account Status Menu Options . . 112
Background About Checking NIS+ Credential Table Setup . . . . . 113
Home . . . . . 113
Why Say Yes to Automatic Creation of Home Directories? 113
Skeleton Path Considerations . . . . . 114
Controlling the Use of Shell Initialization Files . . . . . 114
Labels . . . . . 116
Background on the Clearance and Minimum Label . 116
Background on Displaying Labels . . . . . 117
Background on Showing or Hiding SLs and ILs . . . . 119
Profiles . . . . . 120
Roles . . . . . 122
Idle . . . . . 122

Setting Up or Modifying a User or Role Account . . . . . 124

▼ To Launch the User Manager . . . . . 124

▼ To Load a List of User and Role Accounts Using the Load Dialog Box . . . . . 125

▼ To Load Users or Exit (optional) . . . . . 127

▼ To Find or Sort Accounts . . . . . 128

▼ To Add, Modify or Delete Accounts . . . . . 129

**6. Managing Mail . . . . . 157**

Overview of Trusted Solaris Mail Features. . . . . 158

Multilabel Directories for Outgoing and Incoming Mail . . 159

Mailboxes in Multilabel Directories . . . . . 160

Mail Notification . . . . . 162

Reading of Mail . . . . . 163

How Mail Gets Its Sensitivity Label . . . . . 164

Changing Mail Aliases . . . . . 164

Enabling the Use of `.mailrc` Files in Home Directory MLDs 164

`.copy_files`. . . . . 165

`.link_files`. . . . . 165

Using the `.copy_files` and `.link_files` Along With Skeleton Directories . . . . . 165

▼ To Propagate a `.mailrc` to All Accounts' Home Directory SLDs . . . . . 166

Creating and Initializing New Local and NIS+ Managed Aliases. . . . . 166

▼ To Edit Aliases . . . . . 167

Allowing Users to List the Entire Mail Queue . . . . . 167

▼ To Allow Listing of the Mail Queue . . . . . 168

Tracing Sendmail's Activities . . . . . 168

▼ To Trace Sendmail for Trusted Solaris Information . . . . 170

Troubleshooting Mail Delivery Difficulties . . . . . 171

▼ To Check for a Properly Configured Network Connection for Sending Mail . . . . . 172

Configuring Trusted Solaris Mail Delivery Options for Mail Below Users' Minimum Labels . . . . . 176

How Sendmail Handles Mail Below the Recipient's Minimum SL . . . . . 177

Mail Handling Options . . . . . 177

▼ To Configure Mail Delivery Options for Mail Below Users' Minimum Labels . . . . . 178

Substituting an Alternate Mail Application . . . . . 180

Tip . . . . . 181

▼ To Substitute an Alternate Mail Application in the Front Panel for All Users. . . . . 181

▼ To Create a Multilevel Action for the Alternate Mail Application. . . . . 185

▼ To Install an Alternate Mailer in the Front Panel . . . . . 188

**7. User Manager Data Collection Worksheet . . . . . 191**

User or Role Account Worksheet . . . . . 192

**8. Managing Execution Profiles for Users and Roles . . . . . 193**

Review of Terms. . . . . 194

Execution profiles . . . . . 194

Effective UID and GID . . . . . 195

Actions . . . . . 195
Enabling Attributes . . . . . 196
Restrictive Attributes. . . . . 196
Privileges in Profiles . . . . . 196
Background on Execution Profiles. . . . . 197
Use of the Profile Manager to Create or Modify Execution Profiles 198
Using the Control Buttons on the Profile Manager Dialog Boxes 198
Picking a Naming Service. . . . . 199
Filtering Profiles. . . . . 200
When Adding a New Profile. . . . . 201
When Modifying an Existing Profile . . . . . 201
Launching an Empty Profile Manager. . . . . 201
Launching the Profile Manager Loaded With an Existing Profile. . . . . 202
Bringing Up a Blank Profile Definition, Loading an Existing Profile, or Saving Changes Within the Profile Manager 207
Entering or Changing the Profile Name or Description . . . 208
Switching Among Actions, Commands, and Authorizations Modes . . . . . 209
Working With the Excluded and Included Lists . . . . . 209
Moving Items Between Lists. . . . . 210
Dragging and Dropping Into the Included List . . . . . 211
Moving and Clearing Many List Items With the Select All and Clear All Buttons . . . . . 211

Working With Common Features of the Commands and Actions Modes . . . . . 211

Expanding and Contracting Application Group and Directory Listings in the Command and Actions Modes 211

Using the Buttons to Set Security Attributes on Commands and Actions . . . . . 212

Setting Privileges on Commands and Actions . . . . . 214

Setting a Label Range for a Command or Action . . . . 215

Working in Command Mode . . . . . 217

Loading A New Directory . . . . . 218

Viewing a Command's Man Page . . . . . 219

Working in Authorizations Mode . . . . . 219

Working in Actions Mode . . . . . 221

▼ To Access the Profile Manager . . . . . 223

▼ To Pick a Naming Service and Filter for Profiles . . . . . 224

Specifying a New Profile . . . . . 227

Modifying an Existing Profile . . . . . 228

Execution Profile Procedures . . . . . 229

▼ To Enter the Name and Description for a New Profile . 229

▼ To Specify Commands in the Profile Manager . . . . . 229

▼ To Specify Actions in an Execution Profile . . . . . 230

▼ To Specify Authorizations in an Execution Profile . . . . 232

*Part 3 — Managing Hosts and Networks*

**9. Trusted Solaris Concepts for Managing Hosts and Networks 235**

Trusted Solaris Networking . . . . . 236
Example of a Single Security Domain . . . . . 237
Example of Multiple Security Domains . . . . . 238
Static Routing and Issues About the Use of Trusted Solaris Hosts As Routers . . . . . 238
Network Accreditation Range Requirements. . . . . 239
Static Routing . . . . . 240
**10. Specifying Security Attributes in Trusted Network Databases and Setting Up Routing . . . . . 247**
Goals of Trusted Networking . . . . . 248
How Security Attributes Are Added to Packets . . . . . 249
IP Options . . . . . 249
CIPSO Labels in Packets . . . . . 250
RIPSO Labels in Packets . . . . . 251
MAC Enforcement on Outgoing Messages. . . . . 252
MAC Enforcement on Incoming Messages. . . . . 253
Trusted Network Databases . . . . . 254
Host Types and Trusted Networking . . . . . 255
Host Types Supported . . . . . 256
Trusted Solaris 2.x (`sun_tsol`) Host Type . . . . . 257
Security Attributes Specifiable for the Trusted Solaris 2.5 Host Type . . . . . 257
TSIX (`tsix`) Host Type . . . . . 260
Security Attributes Specifiable for TSIX Hosts. . . . . 260
MSIX (`msix`) Host Type . . . . . 263

CIPSO (cipso) Host Type . . . . . 265
RIPSO (ripso) Host Type . . . . . 267
Unlabeled (unlabeled) Host Type. . . . . 269
Setting Up Trusted Routing . . . . . 271
Creating Entries in the Trusted Networking Databases . . . . . 273
Using tnrhdb Options to Achieve a Closed or Open Type of Network Configuration . . . . . 273
Hierarchical Fallback Mechanism . . . . . 274
Open Configuration Using a Wildcard . . . . . 274
Closely-controlled Configuration. . . . . 274
Setting Up Templates . . . . . 275
Precedence Rules for Attributes in Trusted Network Databases 276
Example of Combined Entries . . . . . 276
Network Accreditation Range . . . . . 279
Procedures . . . . . 280
▼ To Access the Trusted Network Databases from the Database Manager . . . . . 280
▼ To Assign a Template to a Single Host in the tnrhdb . 282
▼ To Assign a Template to a Group of Hosts in the tnrhdb284
▼ To Create a Wildcard tnrhdb Entry for All Hosts Not Otherwise Specified . . . . . 287
▼ To Set an Accreditation Range in a Host Template or Network Interface Entry. . . . . 290
▼ To Configure a Network Interface. . . . . 291
▼ To Add a New Entry or Modify an Existing Entry in tnidb(4TSOL) . . . . . 292

▼ To Substitute a Valid CIPSO Label for the ADMIN_HIGH Sensitivity Label . . . . . 296

▼ To Set Up a Simple Default Route for a Network with One Gateway . . . . . 297

▼ To Set Up Trusted Routing . . . . . 297

**11. Managing Files and File Systems. . . . . 303**

Overview of Files, Directories, and File Systems. . . . . 304

Review of File, Directory, and Filesystem Access Terminology 305

Access Control List . . . . . 305

Access Permissions . . . . . 305

Access Policy for Files, Directories, and File Systems. . . . . 306

Accreditation Range . . . . . 308

Adorned Name . . . . . 309

CMW Label. . . . . 309

Classification . . . . . 309

Clearance . . . . . 309

Compartments . . . . . 309

Discretionary Access Control. . . . . 310

Dominate. . . . . 310

Execution Profile Mechanism. . . . . 310

Information Label . . . . . 310

Information Label Floating. . . . . 311

Label . . . . . 311

Label Range . . . . . 311

Mandatory Access Control. . . . . 311

Markings . . . . . 312
Minimum Label . . . . . 312
Multilevel Directory . . . . . 313
Permission Bits . . . . . 314
Privilege . . . . . 314
Process . . . . . 314
Security Administrator . . . . . 314
Security Attribute . . . . . 315
Security Policy . . . . . 315
Sensitivity Label . . . . . 316
Session Clearance . . . . . 316
Single-level Directory . . . . . 316
Strictly Dominate . . . . . 316
System Accreditation Range . . . . . 317
User Accreditation Range . . . . . 317
User Clearance . . . . . 317
Attributes on Files and Directories . . . . . 318
Changing Security Attributes on Files and Directories . . . . . 320
Changing Labels and Privileges . . . . . 320
Changing File and Directory Attribute Flags . . . . . 321
Attributes on File Systems . . . . . 323
Variable Attribute File Systems . . . . . 323
Specifying Variable Attributes on File Systems . . . . . 324
Fixed Attribute File Systems . . . . . 325

Types of File Systems That May Be Mounted in the Trusted Solaris System . . . . . 327

Specifying Mount Time Security Attributes . . . . . 330

Attribute Precedence Rules. . . . . 331

Example of Specifying Security Attributes for a Fixed Attribute File System Mounted from an Unlabeled Host . . . . . 333

Trusted Solaris NFS Mounts . . . . . 334

Trusted Solaris and NFS . . . . . 334

Exporting Directories for Mounting by Other Hosts. . . . . 335

Troubleshooting Mount Failures . . . . . 336

File and File System-related Procedures . . . . . 336

- To Change Labels and Privileges on Files and Directories 336
- To Specify Alternative Security Attributes While Creating a Local File System . . . . . 339
- To Set Security Attributes on a Standard File System or Reset Security Attributes for an Existing Trusted Solaris File System . . . . . 339
- To Specify Mount-time Security Attributes on the Command Line . . . . . 340
- To Specify Mount-time Security Attributes in the Mount Table . . . . . 340
- To Share a Directory for Mounting by Other Hosts . . . 343
- To Mount a TMPFS-type File System Using the Command Line . . . . . 343
- To Mount a CDROM (HSFS-type File System) Using the Command Line . . . . . 343
- To Trouble Shoot Mount Failures . . . . . 344

**12. Managing NIS+ . . . . . 345**

Managing Multiple Trusted Solaris Hosts in a Security Domain 346

*Managing Standalone* Trusted Solaris Hosts. . . . . 346

NIS+ Constraint on Using the Root Role to Use Solstice System Administration Tools. . . . . 347

New Trusted Solaris NIS+ Tables and Files Not Administered By NIS+. . . . . 347

Adding Trusted NIS+ Tables . . . . . 348

Adding a New Host and Giving It Credentials . . . . . 348

NIS+-Related Procedures . . . . . 349

▼ To Save NIS+ Tables and Restore Them After Reinstalling the Trusted Solaris System . . . . . 349

**13. Changing Configurable Trusted Solaris Kernel Switches . . . 353**

Behaviors Controlled by Configurable Trusted Solaris Kernel Switches . . . . . 354

Needed Terms and Concepts . . . . . 354

`tsol_admin_high_to_cipso` . . . . . 354

`tsol_enable_il` . . . . . 355

tsol_enable_il_floating . . . . . 355

tsol_float_sysv_msg_il . . . . . 355

tsol_float_sysv_sem_il. . . . . 355

tsol_float_sysv_shm_il . . . . . 356

tsol_reset_il_on_exec. . . . . 356

Upgraded Names. . . . . 356

`tsol_hide_upgraded_names`. . . . . 356

`tsol_privs_debug`. . . . . 357

How Kernel Switches Are Set and Changed . . . . . 357

▼ To Change Kernel Switch Setting in the `/etc/system` File 359

Distributing Changed Kernel Switch Settings to Hosts Across the Network . . . . . 361

▼ To Remotely Distribute the `system` File . . . . . 361

**14. Managing Printing . . . . . 363**

Needed Terms. . . . . 364

Banner / Trailer Pages. . . . . 364

Body Pages . . . . . 364

How Trusted Solaris Features Address Information Labeling and Access Control for Printers. . . . . 365

Assigning Labels to Print Jobs . . . . . 366

Using a Label Range on Printers to Control Which Jobs Can Print 367

Printing of Labels on Printer Output. . . . . 368

Default: Information Label . . . . . 368

Alternate Body Page Labeling Options. . . . . 368

Changing the Default Label on Body Pages. . . . . 369

Labels, Job Numbers, and Handling Information Printed on Banner and Trailer Pages . . . . . 370

Changing the Default Labels and Warnings on Print Jobs . 370

`label_encodings`(4TSOL). . . . . 370

`/usr/lib/lp/postscript/tsol_separator.ps` 372

Supported Printers and Types of Output . . . . . 378

Printing PostScript Files . . . . . 378

Printing ASCII and Unsupported File Contents . . . . . 379

Sending Jobs to Printers Connected to Other Print Servers 379

Permitting Publicly-readable Jobs to Be Printed by Default Without Labeled Pages . . . . . 379

Configuring Printers . . . . . 381

Modified Utilities and Man Pages . . . . . 382

Authorizations to Bypass Printing Defaults . . . . . 383

Printing-related Procedures . . . . . 384

- To Access the Printer Manager . . . . . 384
- To Install a Printer on a Print Server . . . . . 386
- To Configure a Restricted Label Range for a Printer Using the Device Manager . . . . . 392
- To Add Access to a Remote Printer . . . . . 394
- To Specify SLs to Print Instead of ILs on Body Pages . . 396
- To Allow Some Users to Print Jobs Without Banners and Trailers . . . . . 397
- To Suppress the Printing of Page Labels on All Print Jobs 398
- To Allow Some Users to Print Jobs Without Page Labels 398
- To Set Up Automatic Printing of Publicly-readable Jobs Without Labels . . . . . 399

**15. Managing Device Allocation and Setting Device Label Ranges 401**

Needed Terms and Concepts . . . . . 402

Security Risks Associated With Use of Devices . . . . . 402

Clearing Objects Prior to Reuse . . . . . 403

How Allocation / Deallocation Make Object Reuse Possible 403
How Manually-allocatable Devices are Allocated and Deallocated and Administered . . . . . 403
How Devices With Removable Media Are Handled . . . . . 406
Controlling the Location of Devices in the Trusted Solaris System . . . . . 407
Device-Clean Scripts . . . . . 407
Device-Clean Script for Tape Devices . . . . . 408
Device-Clean Scripts for Floppy Disks and CD-ROM 408
Device-Clean Script for Audio . . . . . 409
Writing New Device-Clean Scripts . . . . . 409
Label Considerations . . . . . 410
Restricted Label Range . . . . . 410
MAC Rules for Device Access . . . . . 411
Proper Handling of Information Stored on Removable Media 411
Considerations When Importing and Exporting Information 411
Lock Files for Allocatable Devices . . . . . 412
Allocate Error State . . . . . 413
How Non-Allocatable Devices Are Managed . . . . . 413
Default `device_allocate` File Contents and Settings . . . 413
How Device Allocation Works . . . . . 415
Setting Policy for a Device . . . . . 419
Managing Device Allocation and Setting Device Label Ranges 419
Device Management Procedures . . . . . 420

▼ To Add a New Device Allocation Authorization. . . . . 420

▼ To Manage Devices Using the Device Allocation Manager 420

▼ To Add a New Allocatable Device or Make an Existing Device Allocatable. . . . . 423

▼ To Add a New Non-Allocatable Device . . . . . 430

▼ To Change or Add a Device Clean Script . . . . . 433

**16. Adding Software . . . . . 435**

Review of Terms and Concepts . . . . . 437

Controls for Software Creation and Use . . . . . 438

Controls for Importing Software . . . . . 438

Privileges . . . . . 438

Required Privileges . . . . . 439

Override Privileges . . . . . 439

Alternatives to Assigning Privilege. . . . . 440

Principle of Least Privilege. . . . . 440

File Privilege Sets. . . . . 440

How Two Standard Programs Use Privilege in Trusted Solaris 441

Actions . . . . . 441

Effects of the Execution Profiles on the Use of Commands and Actions . . . . . 442

The Profile Shell, the System Shell, and Trusted Processes 443

Processes, Programs, and Their Privileges . . . . . 445

Process Privilege Sets . . . . . 445

Examples of How Processes Acquire Privileges . . . . . 447

In a Standard Shell . . . . . 447
In a Profile Shell . . . . . 448
How a Process Executing the `mount` Command Acquires Privileges . . . . . 449
Why Inheritable Privileges Are Important . . . . . 450
When a Program File Has No Allowed Privileges . . . 450
When a Program File Has No Forced Privileges . . . . 451
How Privileges Are Assigned to Commands and Actions . . . . 453
Giving Forced Privileges to a Command . . . . . 453
Giving Inheritable Privileges to a Command or Action . . . 454
Why Privileged Programs Need to Use Trusted Shared Libraries 454
Default Trusted Shared Library Directories . . . . . 455
Shared Libraries Used by Third Party or Site-Created Applications . . . . . 455
Example of Privileged Java Applications Using Java Libraries 456
Starting Commands With or Without Trusted Solaris Security Attributes During Boot . . . . . 457
Using Scripts in the `/etc/init.d` Directory to Start and Stop Services . . . . . 459
Security Administrator's Tasks in Adding Software . . . . . 460
Issues Around the Adding of Privileges to Any Software . 460
When Adding Existing Programs . . . . . 461
Things to Think About When a Program Fails Without Privileges . . . . . 462
When Applications Need to Be Installed as Root . . . . . 463

When Applications Need to Run As Root . . . . . 464
When Adding a New Trusted Program . . . . . 464
Developer's Responsibilities. . . . . 464
Security Administrator's Responsibilities. . . . . 465
When Adding Actions. . . . . 465
Modifying the Front Panel and Workspace Menu. . . . . 467
Creating and Using Shell Scripts . . . . . 469
Summary of Shell Script Behavior in Trusted Solaris Systems 470
How Program File Are Protected From Being Able to Use Inheritable Privileges If Edited . . . . . 472
Procedures for Adding Software . . . . . 472
▼ To Add A Package from a CD-ROM . . . . . 472
▼ To Set Up an Application to Run WIth a Real UID of Root 473
▼ To Set Up An Application to Run With An Effective UID of Root . . . . . 473
▼ To Find Out Which Privileges an Application Needs . . 473
▼ To Give Forced Privileges to a Command . . . . . 476
▼ To Allow Trusted Programs to Link to Trusted Libraries 478
▼ To Write a Profile Shell Script That Runs Privileged Commands . . . . . 479
▼ To Write a Standard Shell Script That Runs Privileged Commands When Executed in a Profile Shell . . . . . 480
▼ To Specify Commands to Run With or Without Special Attributes During Boot . . . . . 481
▼ To Restore Privileges Lost When a File is Edited. . . . . 483

**17. Host Administration Checklist . . . . . 485**

**A. Profile Summary Tables . . . . . 487**

Execution Profile Content Summary . . . . . 488

Finding Commands in Execution Profiles . . . . . 493

Finding Actions in Execution Profiles . . . . . 506

**B. Profile Definition Tables . . . . . 511**

Profile Names, Purposes, and Roles Tables . . . . . 512

Profile Name: All . . . . . 514

Profile Name: All Authorizations . . . . . 515

Profile Name: Audit Control . . . . . 516

Profile Name: Audit Review . . . . . 519

Profile Name: Basic Actions . . . . . 520

Profile Name: Basic Commands . . . . . 522

Profile Name: Convenient Authorizations . . . . . 525

Profile Name: Enable Login . . . . . 526

Profile Name: Maintenance and Repair . . . . . 527

Profile Name: Media Backup . . . . . 528

Profile Name: Media Restore . . . . . 530

Profile Name: NIS+ Administration . . . . . 532

Profile Name: NIS+ Security Administration . . . . . 533

Profile Name: Nothing . . . . . 534

Profile Name: Object Access Management . . . . . 535

Profile Name: Object Label Management . . . . . 537

Profile Name: Object Privilege Management . . . . . 539

Profile Name: Outside Accred . . . . . 541
Profile Name: Privileged Shells . . . . . 541
Profile Name: System Management. . . . . 542
Profile Name: System Security . . . . . 549
Profile Name: User Management . . . . . 552
Profile Name: User Security . . . . . 553
Profile Name: inetd . . . . . 554
Profile Name: boot . . . . . 557

# *Figures*


Figure 1-1 Disabled Logins Dialog Box for a User Not Authorized to Enable Logins . . . . . 6

Figure 1-2 Disabled Logins Dialog Box for a User Authorized to Enable Logins . . . . . 6

Figure 1-3 Workspace Switch Area with a Button for the admin Administrative Role . . . . . 7

Figure 1-4 Application Manager Icon Selected in the Front Panel, and the System_Admin Folder Selected in the Application Manager. 9

Figure 1-5 Administrative Actions. . . . . 10

Figure 1-6 Disabled Logins Dialog Box for a User Not Authorized to Enable Logins . . . . . 12

Figure 1-7 Disabled Logins Dialog Box for a User Authorized to Enable Logins . . . . . 13

Figure 1-8 Workstation Information Dialog Box . . . . . 14

Figure 1-9 Single Label Indicator on the Message of the Day Dialog Box 15

Figure 1-10 Label Builder Dialog Box for a Single-sensitivity Label Session 17

Figure 1-11 Session Clearance Dialog Box . . . . . 19

Figure 1-12 Choosing the Assume admin Role Option from the Trusted Path Menu . . . . . 20

Figure 1-13 Role Password Dialog Box . . . . . 20
Figure 1-14 Creating a New Role Workspace From the Trusted Path Menu 21
Figure 1-15 A New admin_1 Workspace Button for a New Administrative Role Workspace . . . . . 22
Figure 1-16 Relabeling an Administrative Role Workspace . . . . . 23
Figure 2-1 Format of a `tsolprof` Entry. . . . . 32
Figure 2-2 An Example `tsolprof` Entry . . . . . 32
Figure 2-3 Profiles in a `tsoluser` Entry for an Administrative Account 33
Figure 2-4 TSOL_AUTH Defined Authorizations in `auth_names.h` . . . 34
Figure 2-5 TSOL_AUTH_RESERVED Authorizations in `auth_names.h` 34
Figure 2-6 Authorizations Available for Extension. . . . . 35
Figure 2-7 Format of the `auth_name` File . . . . . 35
Figure 2-8 Definition for the enable logins Authorizations in the `auth_name` File . . . . . 36
Figure 2-9 Specifying a Manifest Constant for a New Authorization in `auth_names.h` . . . . . 36
Figure 2-10 Specifying a Name and a Description For A New Authorization in `auth_name`. . . . . 37
Figure 2-11 Manifest Constants and Numbers for Default Privileges in `priv_names.h` . . . . . 38
Figure 2-12 Privilege Numbers Reserved for Trusted Solaris Use . . . . . 38
Figure 2-13 Privileges Available for Extension . . . . . 39
Figure 2-14 Format of the `priv_name` File . . . . . 39
Figure 2-15 Definition for the file_audit privilege in the `priv_name` File 40
Figure 2-16 Comment from the `priv_names.h` File. . . . . 41
Figure 2-17 Specifying a Manifest Constant for a New Privilege in `priv_names.h` . . . . . 42

Figure 2-18 Specifying a Name and a Description for a New Privilege in `priv_name` . . . . . 42

Figure 2-19 Example `SLD` Name for the Third SLD Created in a Home Directory . . . . . 48

Figure 2-20 Example `SLD` Name for the Fourth SLD Created in a Home Directory . . . . . 49

Figure 2-21 Preparing the File Manger Before Deleting an MLD . . . . . 53

Figure 3-1 User Manager: Navigator . . . . . 63

Figure 3-2 How `$HOME/.dtprofile` is installed . . . . . 66

Figure 3-3 Default Setting in the `/usr/dt/config/sys.dtprofile` . 66

Figure 3-4 How `$HOME/.dtprofile` is Bypassed for Users with a Default Shell of `pfsh(1MTSOL)` . . . . . 68

Figure 3-5 `.mailrc` Example . . . . . 68

Figure 3-6 Startup Files for Commonly Used Applications . . . . . 69

Figure 3-7 Planning Worksheet for Copying and Linking Startup Files Between SLDs . . . . . 72

Figure 3-8 Changing to a Skeleton Directory Created for C Shell Startup Files 84

Figure 3-9 Startup Files in `/etc/skel/skelC` . . . . . 85

Figure 4-1 Division of Account and Profile Configuration Responsibilities Between Security Administrator and System Administrator . 91

Figure 4-2 User Manager: Navigator . . . . . 92

Figure 5-1 User Manager: Navigator . . . . . 100

Figure 5-2 System-wide PAM Settings . . . . . 111

Figure 5-3 Lock-out Password Dialog Box . . . . . 123

Figure 5-4 Launching the User Manager . . . . . 125

Figure 5-5 User Manager: Load Dialog Box with Filter Users Menu . . . . 126

Figure 5-6 User Manager: Main Window and Menus . . . . . 127

Figure 5-7 View Menu with Sort By Submenu ..... 128
Figure 5-8 User Manager: Find Dialog Box ..... 128
Figure 5-9 User Manager Edit Menu ..... 129
Figure 5-10 User Manager: Navigator ..... 130
Figure 5-11 User Manager: Identity Add Dialog Box ..... 132
Figure 5-12 Controls on the User Manager: Identity Dialog Box ..... 133
Figure 5-13 User Manager: Password Dialog Box ..... 134
Figure 5-14 Password Dialog Box: Password Menu ..... 135
Figure 5-15 User Manager: Set Password ..... 135
Figure 5-16 Password Generator Dialog Box ..... 136
Figure 5-17 Password Dialog Box: Password Duration and Expiration Date Fields ..... 138
Figure 5-18 Password Dialog Box: Warning Field ..... 138
Figure 5-19 Password Dialog Box: Generation Field and Menu ..... 139
Figure 5-20 Password Dialog Box: Status Field and Menu ..... 140
Figure 5-21 Credential Table Setup Check Box ..... 140
Figure 5-22 Controls on the User Manager: Password Dialog Box ..... 140
Figure 5-23 User Manager: Home Directory Dialog Box ..... 142
Figure 5-24 Controls on the User Manager: Home Dialog Box ..... 143
Figure 5-25 User Manager: Labels Dialog Box ..... 144
Figure 5-26 Label Builder for Setting the Account's Clearance ..... 145
Figure 5-27 Label Builder for Setting the Minimum SL ..... 147
Figure 5-28 Controls on the User Manager: Labels Dialog Box ..... 148
Figure 5-29 User Manager: Profiles Dialog Box ..... 150
Figure 5-30 Controls on the User Manager: Profiles Dialog Box ..... 151
Figure 5-31 User Manager: Roles Dialog Box ..... 152

Figure 5-32 Controls on the User Manager: Roles Dialog Box . . . . . 153
Figure 5-33 User Manager: Idle Dialog Box with Idle Time Menu . . . . . 154
Figure 5-34 Controls on the User Manager: Idle Dialog Box . . . . . 154
Figure 5-35 Controls on the User Manager Navigator . . . . . 155
Figure 5-36 User Manager: Main Window and File Menu. . . . . 155
Figure 6-1 `/var/spool/mqueue` MLD and its Contents at Different Sensitivity Label. . . . . 160
Figure 6-2 Mailboxes in SLDs at Different Sensitivity Labels . . . . . 161
Figure 6-3 Mail Subpanel With Mail at Multiple Labels . . . . . 162
Figure 6-4 Window Label on a Mail Reader Launched at a Sensitivity Label of INTERNAL_USE_ONLY When Information Labels are Enabled 163
Figure 6-5 Window Label on a Mail Reader Launched at a Sensitivity Label of INTERNAL_USE_ONLY When Information Labels are Disabled 164
Figure 6-6 Sendmail Data Flow Example . . . . . 169
Figure 6-7 OpenWindow's `mailtool` Action Definition from `sunOW.dt` 181
Figure 8-1 Profile Manager: Load Dialog Box . . . . . 198
Figure 8-2 Profile Manager: Load, Naming Service NIS+ . . . . . 199
Figure 8-3 Profile Manager: Load, Naming Service None . . . . . 200
Figure 8-4 Profile Manager: Load, Profile Filter Choices . . . . . 200
Figure 8-5 Choosing None from the Profile Manager: Load, Profile Filter Menu . . . . . 201
Figure 8-6 Empty Profile Manager in Action Mode . . . . . 202
Figure 8-7 Choosing All from the Profile Manager: Load, Filter Profiles Menu 203
Figure 8-8 Profile Manager: Load, Highlighting a Profile Name . . . . . 203
Figure 8-9 Profile Manager With A Profile Loaded . . . . . 204

Figure 8-10 Specifying a Profile to be Loaded in the Profile Manager By Using a Regular Expression . . . . . 205

Figure 8-11 Privileged Shells Profile Listed in the Profile Manager: Open Dialog When P* is Specified in the Filter Profiles Text Field . . . . . 205

Figure 8-12 Profile Manager Loaded With the Privileged Shells Profile . . 206

Figure 8-13 The Profile Manager Profiles Menu For Opening, Saving, and Closing Profiles . . . . . 207

Figure 8-14 Profile Manager: Open, Highlighting a Profile Name . . . . . 208

Figure 8-15 The Profile Name and Description Fields in the Profile Manager 208

Figure 8-16 The Profile Manager View Menu For Switching Between Actions, Commands, and Authorizations . . . . . 209

Figure 8-17 Profile Manager Loaded With the Privileged Shells Profile . . 210

Figure 8-18 Expanding a Grouping Name to List All of Its Contents . . . . 212

Figure 8-19 Buttons for Setting Privileges, Label Range, UID and GID. . . 213

Figure 8-20 Buttons for Setting Privileges, Label Range, UID and GID. . . 213

Figure 8-21 Profile Manager: Set Privileges Dialog Box . . . . . 215

Figure 8-22 Profile Manager: Set Minimum SL Dialog . . . . . 216

Figure 8-23 The Profile Manager Command Mode. . . . . 218

Figure 8-24 Entering the Pathname of the `/etc` Directory to Choose From Its Commands . . . . . 219

Figure 8-25 The Profile Manager in Authorization Mode . . . . . 220

Figure 8-26 Icon and Type in Action Mode . . . . . 221

Figure 8-27 Profile Manager in Action Mode . . . . . 222

Figure 8-28 The Profile Manager Icon Highlighted in the Solstice_Apps Folder 223

Figure 8-29 Profile Manager: Load Dialog Box . . . . . 224

Figure 8-30 Choosing None from the Profile Manager: Load, Naming Service Menu . . . . . 224
Figure 8-31 Profile Manager: Load, Profile Filter Choices . . . . . 225
Figure 8-32 Specifying Profile Names Using a Regular Expression on the Profile Manager: Load, Filter Profiles Menu . . . . . 225
Figure 8-33 The Privileged Shells Profile Displayed When P* is Specified 226
Figure 8-34 Empty Profile Manager in Action Mode . . . . . 227
Figure 9-1 A Single Security Domain . . . . . 237
Figure 9-2 Two Security Domains . . . . . 238
Figure 9-3 Two Security Domains With Differing Accreditation Ranges 240
Figure 9-4 Example of 0 Hops for Communications Between Four Hosts in a Single Security Domain . . . . . 241
Figure 9-5 Example: Default and Network Routes for Two Security Domains with a Single Gateway . . . . . 243
Figure 9-6 Example tsolgateways File for Communications Among Three Networks. . . . . 245
Figure 9-7 Example Complex Gateway Configuration With Routing Tables 246
Figure 10-1 Packet Format. . . . . 249
Figure 10-2 TSIX and Trusted Solaris 2.5 Packet Format . . . . . 249
Figure 10-3 Tnidb Selected in the Database Manager: Load List . . . . . 254
Figure 10-4 Configurable Fields for the sun_tsol Host Type in the Tnrhtp 258
Figure 10-5 Configurable Fields for the tsix Host Type in the Tnrhtp . . . . 261
Figure 10-6 Configurable Fields for the msix Host Type in the Tnrhtp . . . 264
Figure 10-7 Configurable Fields for the cipso Host Type in the Tnrhtp . . 266
Figure 10-8 Configurable Fields for the RIPSO Host Type in the Tnrhtp . 268
Figure 10-9 Configurable Fields for the unlabeled Host Type in the Tnrhtp 270

Figure 10-10 Portions of a Packet Accessible to the Trusted Networking Software . . . . . 272
Figure 10-11 Attribute Precedence Rules . . . . . 276
Figure 10-12 Assigning Some Default Attributes to Communications from Unspecified Hosts . . . . . 277
Figure 10-13 Default Entry for the le0 Interface in the Tnidb Data. . . . . 278
Figure 10-14 Two Network Interfaces and Their Network Accreditation Ranges 279
Figure 10-15 Database Manager Selected in the Solstice_Apps Folder . . . . 280
Figure 10-16 Loading a Naming Service on the Database Manager. . . . . 281
Figure 10-17 Database Manager Selected in the Solstice_Apps Folder . . . . 281
Figure 10-18 Tnidb Selected in the Database Manager: Load List . . . . . 282
Figure 10-19 Adding a Host Entry to Tnrhdb and Specifying a Template . 283
Figure 10-20 Tnrhdb Host Entry Assigned to the Template Named tsol_1 284
Figure 10-21 Adding a Network Entry to Tnrhdb and Specifying a Template 285
Figure 10-22 Tnrhdb Network Entry Assigned to the Template Named tsol 286
Figure 10-23 IP Address and Template Name for a Tnrhdb Fallback Template Entry . . . . . 287
Figure 10-24 Tnrhdb Fallback Template Entry . . . . . 288
Figure 10-25 Assigning Some Default Attributes to Communications from Unspecified Hosts . . . . . 289
Figure 10-26 Default Interfaces Listed in the Tnidb Database. . . . . 292
Figure 10-27 Tnidb Interface le0 Highlighted and the Edit > Modify Option Selected . . . . . 293
Figure 10-28 Interface Manager (Modify) Dialog Box . . . . . 294
Figure 10-29 Add Option Selected from the Tnidb Edit Menu . . . . . 295
Figure 10-30 Interface Manager (Add) Dialog Box . . . . . 296
Figure 10-31 Database Manager: Load List with Tnrhtp Selected . . . . . 298

Figure 10-32 Database Manager: Tnrhtp Database Dialog Box with the tsol_2 Template Name Selected . . . . . 299
Figure 10-33 Trusted Network Template Manager Modify Dialog Box . . . 300
Figure 11-1 File Manager Selected Menu . . . . . 322
Figure 11-2 Attribute Precedence Rules . . . . . 332
Figure 11-3 File Manager Privileges Dialog Box . . . . . 337
Figure 11-4 File Manager Label Builder . . . . . 338
Figure 11-5 Example `vfstab_adjunct` Entries. . . . . 342
Figure 13-1 Label Configuration Defaults . . . . . 359
Figure 14-1 Automatic Labeling of Print Jobs . . . . . 366
Figure 14-2 Example of a Printer with a Restricted Label Range . . . . . 367
Figure 14-3 Information Label Automatically Printed by Default on a Body Page . . . . . 368
Figure 14-4 Sensitivity Label Printed on Body Pages When Information Labels Are Disabled. . . . . 369
Figure 14-5 Typical Print Job Banner Page . . . . . 371
Figure 14-6 Differences on a Trailer Page . . . . . 371
Figure 14-7 Application Manager Icon. . . . . 384
Figure 14-8 Solstice_Apps Icon . . . . . 384
Figure 14-9 Printer Manager Icon Selected in the Solstice_Apps Folder . . 384
Figure 14-10 Printer Manager: Load Dialog Box With None as the Only Naming Service Option . . . . . 385
Figure 14-11 Printer Manager. . . . . 385
Figure 14-12 Application Manager Icon. . . . . 386
Figure 14-13 Solstice_Apps Icon . . . . . 386
Figure 14-14 Serial Manager Icon . . . . . 386

Figure 14-15 Serial Port Manager and Serial Port Manager: Modify Dialog Boxes . . . . . 387

Figure 14-16 Printer Manager: Selecting Install Printer from the Edit Menu 388

Figure 14-17 Printer Manager: Install Printer Dialog Box . . . . . 389

Figure 14-18 Device Allocation: Configuration Dialog Box . . . . . 393

Figure 14-19 Printer Manager: Selecting Add Access to Printer from the Edit Menu . . . . . 394

Figure 14-20 Printer Manager: Add Access to Printer Dialog Box . . . . . 395

Figure 15-1 Device Allocation Manager . . . . . 404

Figure 15-2 Device Allocation, Management, Configuration and Authorizations Tools . . . . . 405

Figure 15-3 The Default device_allocate File . . . . . 414

Figure 15-4 Listing for an Allocatable Device's Lock File . . . . . 415

Figure 15-5 Listing for Device Special Files for an Allocatable Tape Device 416

Figure 15-6 Device Allocate File Entry for mag_tape_0 . . . . . 417

Figure 15-7 Device Special Files Associated With the Tape Device . . . . . 417

ial Files for `mag_tape_0`
Special Files for `mag_tape_0` . . . . . 418

Figure 15-9 Changing Ownership, Group, and Mode of st0's Lock File Upon Allocation . . . . . 419

Figure 15-10 Device Allocation Manager . . . . . 421

Figure 15-11 Device Allocation: Administration Dialog Box . . . . . 422

Figure 15-12 Application Manager Icon . . . . . 423

Figure 15-13 Device Allocation Manager . . . . . 426

Figure 15-14 Device Allocation: Administration Dialog Box . . . . . 427

Figure 15-15 Device Allocation: Configuration Dialog Box . . . . . 428

Figure 15-16 Device Allocation: Configuration Dialog Box . . . . . 429

Figure 15-17 Application Manager Icon. . . . . 430

Figure 15-18 Device Allocation: Configuration Dialog Box. . . . . 433

Figure 16-1 Process Acquiring Forced Privileges When Run in a Normal User's Shell . . . . . 447

Figure 16-2 Process Inheriting Privileges From the Profile Shell . . . . . 448

Figure 16-3 How a Program That Cannot Use Privileges Can Pass Them to A Program That Can . . . . . 451

Figure 16-4 How Forced Privilege Shell Scripts Are Prevented From Passing Forced Privileges to Their Commands. . . . . 452

Figure 16-5 `/etc/initd.d/sendmail` Linked to `/etc/rc`*n*`.d` Directories 457

Figure 16-6 Starting and Stopping sendmail Using the `start` and `stop` Options with the `/etc/init.d/sendmail` Script. . . . . 459

Figure 16-7 How Shell Scripts Invoked in `pfsh` Can Pass Inheritable Privileges to Their Commands. . . . . 471

Figure 16-8 Changing the Value of `tsol_privs_debug` to `= 1` in the `system` File . . . . . 473

Figure 16-9 Commented Privilege Debugging Line in `/etc/syslog.conf` 474

Figure 16-10 Privilege Debugging Line with `local0.debug` String Added in `/etc/syslog.conf` . . . . . 474

Figure 16-11 `runpd` Displaying Privilege Needed For A Process To Succeed 475

Figure 16-12 Typical Privilege Debugging Entry in `/var/log/privedebug.log`. . . . . 476

# *Tables*


Table 1-1 Administrative Actions, Purposes, and Default Roles. . . . . 10

Table 2-1 MLD-related Commands and What They Do. . . . . 47

Table 3-1 Authorizations for User Manager Buttons and Types of Information Specified . . . . . 64

Table 3-2 Startup Files Read by the Window System . . . . . 65

Table 3-3 crontab(1TSOL) Options. . . . . 80

Table 3-4 Trusted Solaris 2.5 `at`(1) options . . . . . 81

Table 3-5 Trusted Solaris 2.5 `atq`(1) Changes . . . . . 81

Table 3-6 Trusted Solaris 2.5 `atrm`(1) Changes. . . . . 82

Table 4-1 Commands and Applications Requiring the Trusted Path Attribute 90

Table 4-2 Authorizations For Specifying Types of User Information. . . 93

Table 5-1 Password Creation Options, Descriptions and Recommendations 108

Table 5-2 Password Rules for Manually Created Passwords. . . . . 109

Table 5-3 Status Menu Options, Descriptions, and Recommendations . 112

Table 5-4 Defaults You Can Set on the User Manager Dialog Boxes . . . 131

Table 6-1 Trusted Solaris Mail Handling Options . . . . . 178

Table 9-1 Example tnidb Entry to Restrict a Network's Accreditation Range 240

Table 10-1 Supported Classifications for RIPSO Labels . . . . . 251

Table 10-2 Protection Authority Flags That May Be Specified in the RIPSO Send PAF or as RIPSO Return PAF Fields. . . . . 252

Table 10-3 Supported Host Types . . . . . 256

Table 10-4 Required and Optional Security Attributes for sun_tsol Host Types 259

Table 10-5 Required and Optional Security Attributes for tsix Host Types 262

Table 10-6 Default tnidb Settings . . . . . 291

Table 11-1 File and Directory Attributes from the Base and from Trusted Solaris Operating System . . . . . 318

Table 11-2 Security Attributes on Files and Directories . . . . . 318

Table 11-3 Trusted Solaris File System Security Attributes With Defined Settings . . . . . 323

Table 11-4 Attributes for Fixed File Systems with Default Values. . . . . 326

Table 11-5 Mount Types, Examples, and Notes. . . . . 328

Table 12-1 New NIS+ Tables . . . . . 347

Table 12-2 Configuration Files Not Administered Through NIS+. . . . . 348

Table 12-3 nisscript for Dumping NIS+ Tables into ASCII Files . . . . . 349

Table 13-1 Default Kernel Switch Settings and Values Definitions. . . . . 357

Table 14-1 `tsol_separator.ps` Comments and Configurable Values 372

Table 14-2 Authorizations Related to Printing. . . . . 383

Table 15-1 Device-Clean Scripts for the Three Supported Tape Devices . 408

Table 15-2 Device-Clean Scripts for Floppy and CD-ROM . . . . . 408

Table 15-3 Device-Clean Script for Audiotool . . . . . 409

Table 15-4 Required Lock File Characteristics for Allocatable and Allocated Devices . . . . . 412
Table 15-5 Lock File Settings for Device in the Allocate Error State . . . . . 413
Table 15-6 Default Settings the `device_allocate` File . . . . . 414
Table 16-1 Example of an Incorrect Ordering of Profiles in an Account's List 443
Table 16-2 Default Directories for Trusted Shared Libraries . . . . . 455
Table 16-3 Differences in Creating and Using Actions Under Trusted Solaris Restraints . . . . . 466
Table 16-4 Differences in Modifying the Workspace Under Trusted Solaris Restraints . . . . . 468
Table 16-5 Default Directories for Trusted Shared Libraries . . . . . 478
Table 17-1 Host Administration Checklist . . . . . 485
Table A-1 Execution Profile Contents . . . . . 488
Table A-2 Commands and Their Associated Execution Profiles . . . . . 493
Table A-3 Actions and Their Associated Execution Profiles . . . . . 506
Table B-1 Name and Purpose of Profiles . . . . . 512
Table B-2 Profiles Assigned to the Default Roles . . . . . 513
Table B-3 Common Profiles (in More Than One Role . . . . . 514
Table B-4 Profiles That Divide Responsibilities Among Roles . . . . . 514
Table B-5 Profiles Unique to One Role . . . . . 514
Table B-6 Actions Defined for the All Profile . . . . . 515
Table B-7 Authorizations Defined for the All Profile . . . . . 515
Table B-8 Commands Defined for the All Profile . . . . . 515
Table B-9 Actions Defined for the All Authorizations Profile . . . . . 515
Table B-10 Authorizations Defined for the All Authorizations Profile . . . 515
Table B-11 Commands Defined for the All Authorizations Profile . . . . . 516

Table B-12 Actions Defined for the Audit Control Profile . . . . . 516
Table B-13 Authorizations Defined for the Audit Control Profile. . . . . 517
Table B-14 Commands Defined for the Audit Control Profile. . . . . 517
Table B-15 Actions Defined for the Audit Review Profile . . . . . 519
Table B-16 Authorizations Defined for the Audit Review Profile. . . . . 520
Table B-17 Commands Defined for the Audit Review Profile . . . . . 520
Table B-18 Actions Defined for the Basic Actions Profile . . . . . 520
Table B-19 Authorizations Defined for the Basic Actions Profile . . . . . 521
Table B-20 Commands Defined for the Basic Actions Profile . . . . . 521
Table B-21 Actions Defined for the Basic Commands Profile . . . . . 522
Table B-22 Authorizations Defined for the Basic Commands Profile. . . . 522
Table B-23 Commands Defined for the Basic Commands Profile . . . . . 522
Table B-24 Actions Defined for the Convenient Authorizations Profile. . 525
Table B-25 Authorizations Defined for the Convenient Authorizations Profile 526
Table B-26 Commands Defined for the Convenient Authorizations Profile 526
Table B-27 Actions Defined for the Enable Login Profile . . . . . 526
Table B-28 Authorizations Defined for the Enable Login Profile . . . . . 526
Table B-29 Commands Defined for the Enable Login Profile. . . . . 527
Table B-30 Actions Defined for the Maintenance and Repair Profile . . . . 527
Table B-31 Authorizations Defined for the Maintenance and Repair Profile 527
Table B-32 Commands Defined for the Maintenance and Repair Profile. 527
Table B-33 Actions Defined for the Media Backup Profile . . . . . 529
Table B-34 Authorizations Defined for the Media Backup Profile . . . . . 529
Table B-35 Commands Defined for the Media Backup Profile . . . . . 529
Table B-36 Actions Defined for the Media Restore Profile . . . . . 530

Table B-37 Authorizations Defined for the Media Restore Profile . . . . . 530
Table B-38 Commands Defined for the Media Restore Profile . . . . . 531
Table B-39 Actions Defined for the NIS+ Administration Profile . . . . . 532
Table B-40 Authorizations Defined for the NIS+ Administration Profile 532
Table B-41 Commands Defined for the NIS+ Administration Profile . . . 532
Table B-42 Actions Defined for the NIS+ Security Administration Profile 533
Table B-43 Authorizations Defined for the NIS+ Security Administration Profile . . . . . 533
Table B-44 Commands Defined for the NIS+ Security Administration Profile 533
Table B-45 Actions Defined for the Enable Login Profile . . . . . 535
Table B-46 Authorizations Defined for the Enable Login Profile . . . . . 535
Table B-47 Commands Defined for the Enable Login Profile. . . . . 535
Table B-48 Actions Defined for the Object Access Management Profile . 535
Table B-49 Authorizations Defined for the Object Access Management Profile 536
Table B-50 Commands Defined for the Object Access Management Profile 536
Table B-51 Actions Defined for the Object Label Management Profile. . . 537
Table B-52 Authorizations Defined for the Object Label Management Profile 538
Table B-53 Commands Defined for the Object Label Management Profile 538
Table B-54 Actions Defined for the Object Privilege Management Profile 539
Table B-55 Authorizations Defined for the Object Privilege Management Profile . . . . . 540
Table B-56 Commands Defined for the Object Privilege Management Profile 540
Table B-57 Actions Defined for the Outside Accred Profile . . . . . 541
Table B-58 Authorizations Defined for the Outside Accred Profile . . . . . 541

Table B-59 Commands Defined for the Outside Accred Profile ..... 541
Table B-60 Actions Defined for the Privileged Shells Profile ..... 542
Table B-61 Authorizations Defined for the Privileged Shells Profile .... 542
Table B-62 Commands Defined for the Privileged Shells Profile ..... 542
Table B-63 Actions Defined for the System Management Profile ..... 542
Table B-64 Authorizations Defined for the System Management Profile. 543
Table B-65 Commands Defined for the System Management Profile.... 544
Table B-66 Actions Defined for the System Security Profile..... 549
Table B-67 Authorizations Defined for the System Security Profile ..... 550
Table B-68 Commands Defined for the System Security Profile ..... 550
Table B-69 Actions Defined for the User Management Profile ..... 552
Table B-70 Authorizations Defined for the User Management Profile... 552
Table B-71 Commands Defined for the User Management Profile ..... 553
Table B-72 Actions Defined for the User Security Profile ..... 553
Table B-73 Authorizations Defined for the User Security Profile ..... 553
Table B-74 Commands Defined for the User Security Profile ..... 554
Table B-75 Actions Defined for the inetd Profile ..... 554
Table B-76 Authorizations Defined for the inetd Profile..... 554
Table B-77 Commands Defined for the inetd Profile..... 555
Table B-78 Actions Defined for the boot Profile..... 557
Table B-79 Authorizations Defined for the boot Profile ..... 558
Table B-80 Commands Defined for the boot Profile ..... 559

# *Preface*

This *Trusted Solaris Administrator's Procedures* manual provides procedures for managing users and machines while maintaining the security of information within the Trusted Solaris™ environment. This book is part of the Trusted Solaris administrator's document set, which also includes an administrative overview, a guide to planning, administering, and configuring the system, and manuals on administering auditing and administering labels.

## *Who Should Use This Book*

This book is used by administrators who are authorized 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™ 2.x administration, CDE, Solstice™, and NIS+**
  The procedures described here are unique to Trusted Solaris administration. Administrators of Trusted Solaris must already understand how to work within and administer the Solaris 2.x operating environment, which is the operating system upon which Trusted Solaris is based, and understand how to use and administer the Common Desktop Environment (CDE) window system, Solstice™ AdminSuite™ system administration tools and NIS+.

- **Read and understand the basic concepts and procedures for using the system, as described in the Trusted Solaris user document set**
  Administrators should understand how to work in the Trusted Solaris environment as a normal user.
- **Read and understand the administrative concepts described in the Trusted Solaris administrator's overview manual**
- **Understand how administrative tasks are divided among roles at your site**
  Each task includes a note that identifies which role has been assigned to the task in the default configuration. Reference tables in Appendix A list the responsibilities, the relevant actions and commands for each of the default roles. Your local security administrator is responsible for making all administrators aware of the new configuration if the default administrative roles have been reconfigured at your site.

## *How This Book Is Organized*

This book has three parts and a total of 18 chapters.

**Part 1 — "Procedures Common to All Tasks and Administrative Roles"**

**Chapter 1, "Assuming a Role and Working in a Role Workspace"**

Reviews how to log in, enable logins, assume an administrative role, work in an administrative role workspace, launch administrative actions from the workspace and administrative commands in the profile shell.

**Chapter 2, "Miscellaneous Tasks and Procedures"**

Describes how to get hexadecimal equivalents for labels and clearances, extend extendable mechanisms, including adding authorizations and privileges, with cross-references to the procedures for adding execution profiles and audit events, and how to administer multilevel directories.

**Part 2 — "Administering Users, Roles, Profiles and Mail"**

**Chapter 3, "Managing User Accounts"**

Describes how the responsibilities for managing user accounts are divided between two administrative roles, what decisions to make and what to do before setting up accounts for users and roles, and how to administer startup files in the Trusted Solaris system.

**Chapter 4, “Managing Roles”**

Describes the differences between user and role accounts, how the responsibilities for managing all roles are given to the security administrator, except for the responsibility of managing the security administrator’s own role, which is given to the system administrator. Describes how the User Manager is used in administering role accounts and introduces the profile manager that is used to configure the profiles for new roles or to modify the profiles for existing ones.

**Chapter 5, “Using the User Manager to Configure User and Role Accounts”**

Describes the information that must be provided in each of the dialog boxes of the User Manager, then provides step by step procedures for entering the required information in the fields.

**Chapter 6, “Managing Mail”**

Describes the differences between standard Solaris and Trusted Solaris mail administration, including the new Trusted Solaris `sendmail` debugging options and the new privacy options in the `sendmail.cf` file for handling mail that is received below an account’s minimum label.

**Chapter 7, “User Manager Data Collection Worksheet”**

Provides a checklist of all the tasks for setting up a new user account.

**Chapter 8, “Managing Execution Profiles for Users and Roles”**

Describes how to use the Profile Manager to create, modify and delete profiles. This chapter includes tables listing the actions, commands and command privileges assigned to each of the default role and user profiles.

**Part 3 — “Managing Hosts and Networks”**

**Chapter 9, “Trusted Solaris Concepts for Managing Hosts and Networks”**

Reviews concepts that apply to managing communications and illustrates how trusted communications can be configured between the Trusted Solaris distributed system and multiple networks.

**Chapter 10, “Specifying Security Attributes in Trusted Network Databases and Setting Up Routing”**

Describes how to specify the security attributes and set up routing for trusted network communications between machines.

**Chapter 11, “Managing Files and File Systems”**

Describes the new file system security attributes and how to set up mounts across the distributed system and specify the new security attributes.

**Chapter 12, “Managing NIS+”**

Describes how NIS+ is used to centrally administer the Trusted Solaris distributed system and lists the new NIS+ tables.

**Chapter 13, “Changing Configurable Trusted Solaris Kernel Switches”**

Describes how to modify the system configuration switches that set whether ILs are displayed, whether ILs float, whether ILs are reset when a new program is executed with `execve(2TSOL)`, whether the names of upgraded files and directories are displayed, and whether privilege debugging mode is turned on.

**Chapter 14, “Managing Printing”**

Describes how to configure printing, how to add and delete labeled and unlabeled printers, and provides pointers to the manual that describes how to specify handling caveats for printer banner and trailer pages.

**Chapter 15, “Managing Device Allocation and Setting Device Label Ranges”**

Describes how to set up device allocation, how to make devices allocatable, how to write and specify device clean scripts to run when a device is deallocated and how to respond to the *allocate error state*. This chapter also includes how to set the label range on devices.

**Chapter 16, “Adding Software”**

Describes how to add Sun unbundled products, UNIX applications, new trusted programs, actions, and shell scripts, how to assess what privileges new software needs and whether to give it the needed privileges. This chapter describes the two ways that privileges are made available to software.

**Chapter 17, “Host Administration Checklist”**

Provides a worksheet for collecting information for setting up a host.

**Appendix A, “Profile Summary Tables”**

Summarizes the contents of the default set of execution profiles.

**Appendix B, “Profile Definition Tables”**

Lists for each profile all of its authorizations, commands and actions and the privileges and security attributes associated with the commands and actions.

## *Related Books*

- *Trusted Solaris User’s Guide*

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

- *Trusted Solaris Administration Overview*
- *Trusted Solaris Administrator’s Procedures*
- *Trusted Solaris Label Administration*
- *Trusted Solaris Audit Administration*
- *Trusted Solaris Developer’s Guide*
- *Trusted Solaris Reference Manual*
- *Trusted Solaris 2.5 Release Notes*
- *Trusted Solaris 2.5 Transition Guide*
- *Compartmented Mode Workstation Labeling: Encodings Format*

## *What Type Styles Mean in Text and Examples*

Table P-1 explains the type styles used in this manual.

*Table P-1* Typographic Conventions

| Type Style | Meaning | Example |
|---|---|---|
| `Filename\|Command` | The names of commands, files, and directories | Edit your `.login` file.<br>Use `ls -a` to list all files. |
| ScreenText | Onscreen computer output | `machine_name%`<br>`You have mail.` |
| **`UserType`** | What you type, contrasted with on-screen computer output | `machine_name%` **`su`**<br>`Password:` |

*Table P-1* Typographic Conventions

| Type Style | Meaning | Example |
|---|---|---|
| *Variable* | Argument name in a command-line. | To delete a file, enter `rm` *filename*. |
| | You replace the argument with a real name or value. | machine_name% **`rm myfile`** |
| *Title* or *Emphasis* | Book titles, new words or terms, or words to be emphasized | Read Chapter 6 in *User's Guide*. These are called *class* options. |
| | | You *must* be root to do this. |

## *Which Trusted Solaris Prompts Go With Which User Types*

Table P-2 identifies which prompts go with which shells.

*Table P-2* Trusted Solaris Prompts

| Shell | Prompt |
|---|---|
| C shell prompt | hostname% |
| Bourne shell, Korn shell and Profile shell prompt | $ |

# *Part 1 — Procedures Common to All Tasks and Administrative Roles*

This part of the *Trusted Solaris Administrator's Procedures* manual contains two chapters that describe the tasks and procedures common to all roles.

Chapter 1, "Assuming a Role and Working in a Role Workspace" includes these topics:


| |
|---|
| *Review of Administrative and Non-administrative Role Concepts* |
| *Working in the Administrative Role Workspace* |
| *To Launch Administrative Applications* |
| *To Launch Administrative Actions* |
| *Preventing Logins From Being Disabled After a Reboot* |


Chapter 2, "Miscellaneous Tasks and Procedures" includes these topics:


| |
|---|
| *Getting a Hexadecimal Equivalent for Labels and Clearances* |
| *Extending Extendable Security Mechanisms* |
| *To Add An Authorization* |
| *To Add a Privilege* |
| *Changing the Trusted Stripe* |
| *Working with MLDs* |

# *Assuming a Role and Working in a Role Workspace* 1

As described in the *Trusted Solaris Administration Overview*, administrative tasks in the Trusted Solaris system are performed by administrators who first log in using their own user account names and passwords, and who, after identifying themselves to the system, then assume a role. This chapter describes the sequence for assuming a role and provides guidance on how to work in administrative role workspaces. This chapter covers the following major topics:


| | |
|---|---|
| *Review of Administrative and Non-administrative Role Concepts* | *page 4* |
| *Working in the Administrative Role Workspace* | *page 7* |


This chapter provides the procedures listed here:


| | |
|---|---|
| *To Log In and Assume an Administrative Role* | *page 12* |
| *To Switch Among Administrative Role Workspaces and the Normal User Workspaces* | *page 21* |
| *To Work at Multiple Labels While in an Administrative Role* | *page 21* |
| *To Launch Administrative Applications* | *page 23* |
| *To Launch Administrative Actions* | *page 24* |
| *To Create A New Administrative Action for Editing an Administrative File* | *page 24* |
| *To Add Actions Outside the System_Admin Folder* | *page 26* |
| *To Prevent Logins From Being Disabled After a Reboot* | *page 26* |

# 1

# Review of Administrative and Non-administrative Role Concepts

The following sections review some concepts introduced in the *Trusted Solaris User's Guide* for convenience, so you can see the complete process from the point of view of someone logging in to do administrative work. Trusted Solaris has two types of roles: *administrative roles* and *non-administrative roles*.

## Administrative Roles

Most administrative work is done administrative roles.

Administrative roles are similar in most ways to non-administrative roles, except that:

- Administrative roles are in sysadmin group 14, which is necessary for performing NIS+-related tasks, and
- Administrative roles work in a special *administrative role workspace* whose processes have the trusted path attribute that is required by many programs used to perform administrative tasks.

Most tasks are divided between the security administrator role and the system administrator role, both of which are administrative roles. A third administrative role, the root role, is used to install software that requires a real UID of root in order to succeed.

## Non-administrative Roles

In the default system, the only non-administrative role used to do administration work is the operator role, which is assigned the backing up of file systems.

## Logging In and Assuming a Role

In the Trusted Solaris system, administrators do not log in directly as administrators. Following is the sequence of how roles of both types log in and begin work in the window system (which is the same sequence for assuming non-administrative roles except for the final bullet):

- The security administrator assigns an administrative or non-administrative role to the user's account, using the Trusted Solaris version of the Solstice User Manager.
- The security administrator gives to the user both the password for the user account and the password for the role.
- The user who is configured to be able to assume a role logs in by supplying his or her own username and password.
- To begin work in the role, the user assumes the role by choosing the assume role option from the Trusted Path menu, and by providing the role password to the role password dialog box.
- For administrative roles, a special *administrative role workspac*e becomes active, and the person then may act in the administrative role to perform administrative tasks.
- For non-administrative roles, a normal user workspace is created that does not have the trusted path attribute that is checked by many administrative programs.

## *Auditing of Administrative Actions*

At login, a UID that also serves as an audit ID becomes associated with all processes generated by the logged-in user. If a user assumes a role, the user's effective ID changes, but the audit ID does not. The audit ID makes it possible to trace back to an identified user account the actions that a user performs while in a role. Requiring users to log in and identify themselves before assuming a role plugs one possible security hole that can occur in a normal UNIX system, in which any individual who discovers the root password can log in directly and have unlimited anonymous access to all the information on the system.

## *How Logins are Enabled*

Logins are disabled by default after every reboot. When logins are disabled, only a user account with the *enable logins* authorization can log in or enable logins. The purpose of disabling logins is to make it possible for the system administrator or other trusted employee to verify that the security of the system has not been compromised before allowing anyone else to log in again.

Any employee assigned the system administrator role should have the enable logins authorization assigned in an execution profile.

If the system has been rebooted and logins have not yet been enabled, one of two dialog boxes displays after the user enters his or her password. If the current user account is not authorized to enable logins, the dialog box shown in Figure 1-1 displays.

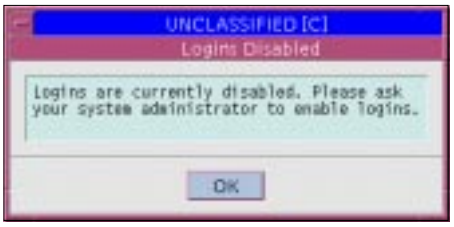



*Figure 1-1* Disabled Logins Dialog Box for a User Not Authorized to Enable Logins

If the account logging in is authorized to enable logins, the dialog box shown in Figure 1-9 displays.

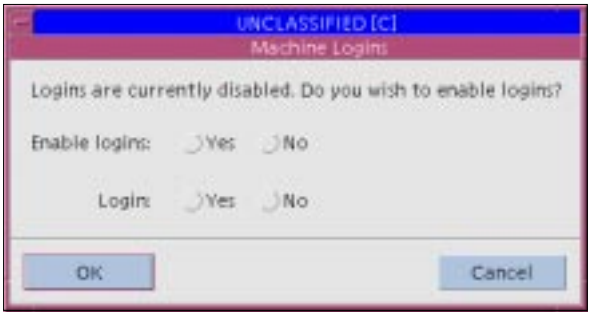



*Figure 1-2* Disabled Logins Dialog Box for a User Authorized to Enable Logins

## *Preventing Logins From Being Disabled After a Reboot*

The presence of the `/etc/nologin` file is checked during the login sequence after a host is rebooted, and logins are disabled if the `nologin` file is present. If it is consistent with the site's security policy to allow logins by everyone after all reboots, the security administrator can edit the `S05RMTMPFILES` script

in /etc/rc2.d to comment out the lines that create the /etc/nologin file. See “To Prevent Logins From Being Disabled After a Reboot” on page 26, if changing the default is consistent with your site’s security policy.

Before removing the lines that create the nologin file, save a copy of S05RMTMPFILES, making sure that the copy has the original files’ DAC and MAC attributes in case you wish later to restore it. The S05RMTMPFILES file's attributes to be maintained are ADMIN_LOW[ADMIN_LOW], owner root and group sys.

## *Working in the Administrative Role Workspace*

Immediately after a user assumes an administrative role for the first time, as described in “To Log In and Assume an Administrative Role” on page 12, an administrative role workspace becomes active at the role’s minimum label, ADMIN_LOW, and a new workspace button for the role displays in the workspace switch area. The new role workspace button is highlighted along the edge of the left top and side and given the name of the administrative role account, as shown in Figure 1-3. The administrative role can change the name of the button just as anyone can change the name of any other workspace.

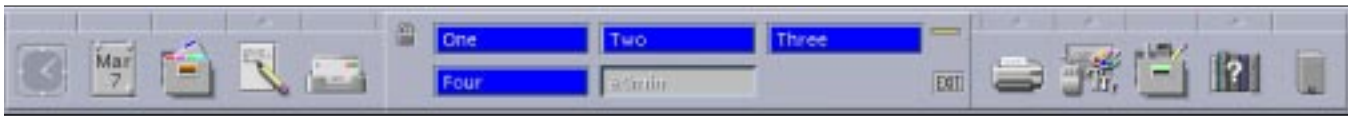



*Figure 1-3* Workspace Switch Area with a Button for the admin Administrative Role

The user who has assumed a role may return to any non-administrative role workspace at any time to work with the commands and actions specified in that account’s execution profile. See “To Switch Among Administrative Role Workspaces and the Normal User Workspaces” on page 21.

Administrative roles most often work at the ADMIN_LOW and ADMIN_HIGH administrative labels. After the first workspace comes up at ADMIN_LOW, if you want to create a new workspace and relabel it for working at ADMIN_HIGH or other sensitivity label, follow the instructions under “To Work at Multiple Labels While in an Administrative Role” on page 21.

Working at more than one label while you are in a role is similar to working at more than one label as a normal user, but bringing up a new administrative role workspace is slightly different. See “To Bring Up New Role Workspaces and Relabel Them.”

## *Application Manager Folders and Actions Icons*

Much of the administrative work performed by administrative roles is done by launching administrative programs from folders within the Application Manager. In keeping with the principle of least privilege, the set of administrative programs is divided between the two major administrative roles. If a role is not allowed to use an application, its icon does not display.

## *Using Administrative Applications in Solstice_Apps Applications Folder*

Administrative roles do most administrative work by launching administrative applications from the Solstice_Apps folder within the Application Manager. See “To Launch Administrative Applications” on page 23. All configuration files that are managed as NIS+ maps are edited through the applications in the Solstice_Apps folder.

## *Using Administrative Actions in the System_Admin Applications Folder*

Other tasks performed by administrative roles, such as the editing of most configuration files that are not managed through NIS+, are done by using the administrative actions that are found in Trusted Solaris version of the System Administration Folder in the Applications Manager.

## *Accessing the Application Manager*

The Application Manager is accessed by clicking its icon. Figure 1-4 shows the Application Manager icon selected in the mid-right side of the Front Panel.

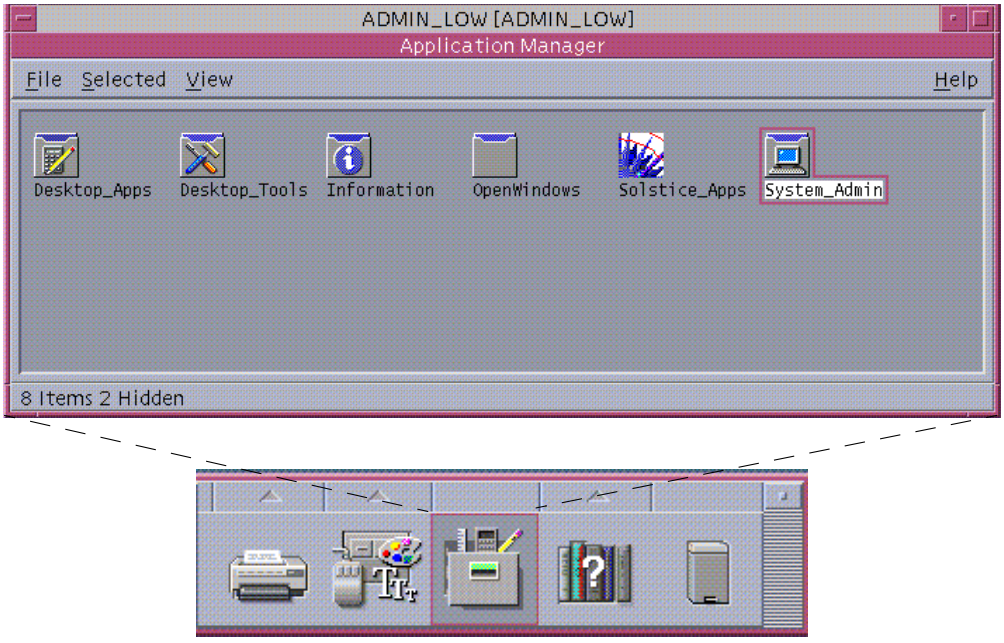



*Figure 1-4* Application Manager Icon Selected in the Front Panel, and the System_Admin Folder Selected in the Application Manager

Figure 1-5 shows the administrative actions that are in the System_Admin folder when it is accessed by the security administrator.

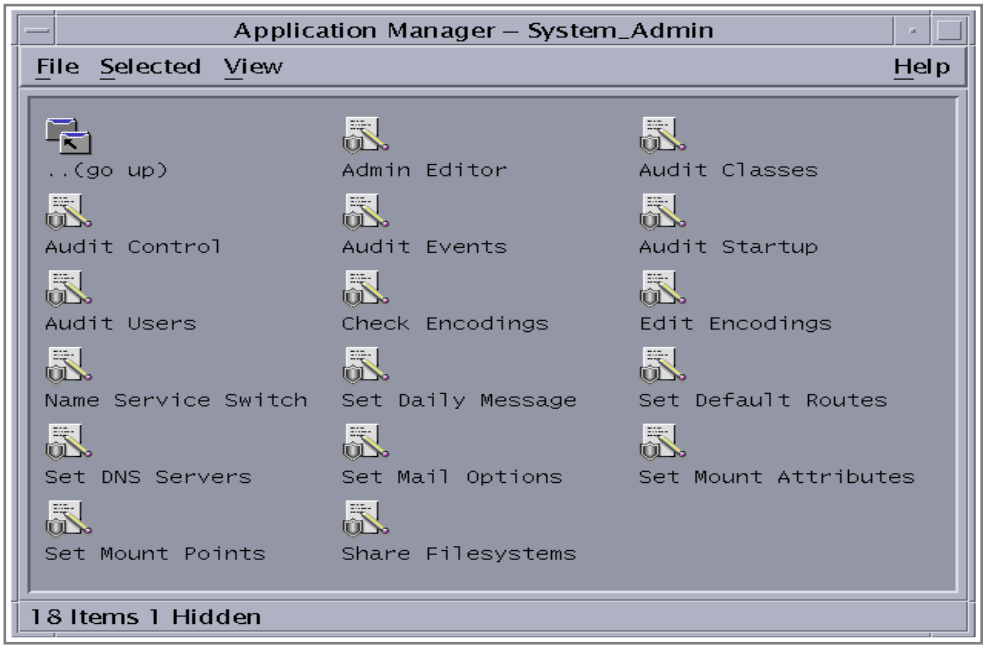



*Figure 1-5* Administrative Actions

*Table 1-1* Administrative Actions, Purposes, and Default Roles

| Action Name | Action Purpose | Default Role |
|---|---|---|
| Admin Editor | Edits any specified file | secadmin |
| Audit Classes | Edits `audit_class(4TSOL)` | secadmin |
| Audit Control | Edits `audit_control(4TSOL)` | secadmin |
| Audit Events | Edits `audit_event(4TSOL)` | secadmin |
| Audit Startup | Edits the `audit_startup.sh` script [see `audit_startup(1MTSOL)`] | secadmin |
| Audit Users | Edits `audit_user(4TSOL)` | secadmin |
| Check Encodings | Runs `chk_encodings(1MTSOL)` on specified encodings file | secadmin |
| Edit Encodings | Edits specified `label_encodings(4TSOL)` file and runs `chk_encodings(1MTSOL)` | secadmin |
| Name Service Switch | Edits `nsswitch.conf(4)` | secadmin |

*Table 1-1* Administrative Actions, Purposes, and Default Roles

| Action Name | Action Purpose | Default Role |
|---|---|---|
| Set Daily Message | Edits `/etc/motd` | admin |
| Set Default Routes | Edits `/etc/defaultrouter` | admin |
| Set DNS Server | Edits `resolv.conf(4)` | admin |
| Set Mail Options | Edits `/etc/mail/sendmail.cf` | secadmin |
| Set Mount Attributes | Edits `vfstab_adjunct(4TSOL)` | secadmin |
| Set Mount Points | Edits `vfstab(4)` | admin |
| Share Filesystems | Edits `dfstab(4)`; does not run `share(1M)` | admin |

See “To Launch Administrative Actions” on page 24.

All the actions are wrappers around the `/usr/dt/bin/trusted_edit` shell script that brings up the `adminvi` restricted editor and audits any changes made at the time the file is saved. [See “Administrative vi” on page 12 and the man page for the characteristics of `adminvi(1MTSOL)`.]

## *Accessing Commands and Actions*

While in the administrative role workspace, the person acting in the role has access only to the commands and actions with the attributes specified for them in the role’s execution profile or profiles. The application or action is launched at the sensitivity label of the currently active role workspace.

## *Using the Profile Shell To Do Other Tasks*

The profile shell, `pfsh(1MTSOL)`, is the default shell for administrative roles. It is used in Trusted Solaris administration to restrict each role to the minimum set of commands that the role needs for doing its tasks, while allowing the commands to inherit any privileges they need while executing. Profile shells may be run at any sensitivity label. Administrators access the profile shell by bringing up dtterm or any other terminal emulator from the front panel, and the profile shell is started in the terminal at the sensitivity label of the current workspace.

### *Administrative* `vi`

The `adminvi(1MTSOL)` command is a modified version of the `vi` command that is restricted to prevent the user from executing shell commands and from writing to (saving to) any file other than the original file being edited. The Admin Edit action, which is assigned to the security administrator role by default, should be used in most cases instead of `adminvi` on the command line to edit or create administrative files. The `adminvi` command may be assigned to any users whose default shell is the profile shell, if the security administrator wishes to allow those users the use of a text editor while constraining them to edit files with adminvi's built-in restrictions.

## *Administrative Role Procedures*

### ▼ To Log In and Assume an Administrative Role

1. **Log in as a normal user, supplying your own username and password.**

   If you enter your username and password and the workstation information dialog box displays as shown in Figure 1-6 on page 12, go to Step 3 on page 15.

2. **Enable logins if necessary (see Figure 1-6 and Figure 1-7).**

   a. **If your account is not authorized to enable logins, notify the security administrator.**
   The security administrator must verify that the security of the system has not been breached before enabling logins so that you can log in. See Figure 1-6.

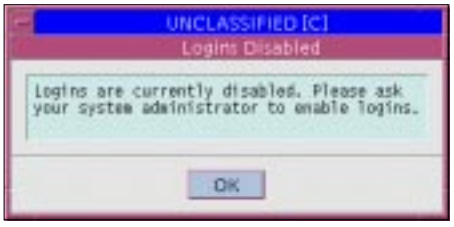



*Figure 1-6* Disabled Logins Dialog Box for a User Not Authorized to Enable Logins

**b. If your account is authorized to enable logins, click the Enable Logins radio button to enable logins.**
See Figure 1-7.

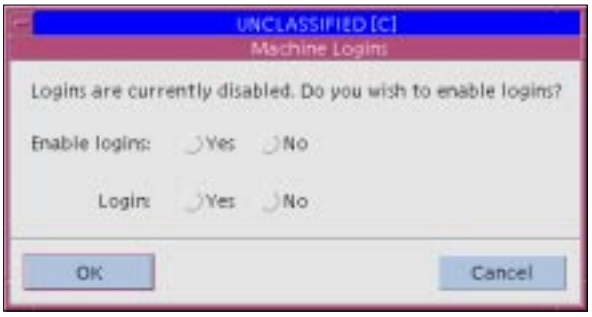



*Figure 1-7* Disabled Logins Dialog Box for a User Authorized to Enable Logins

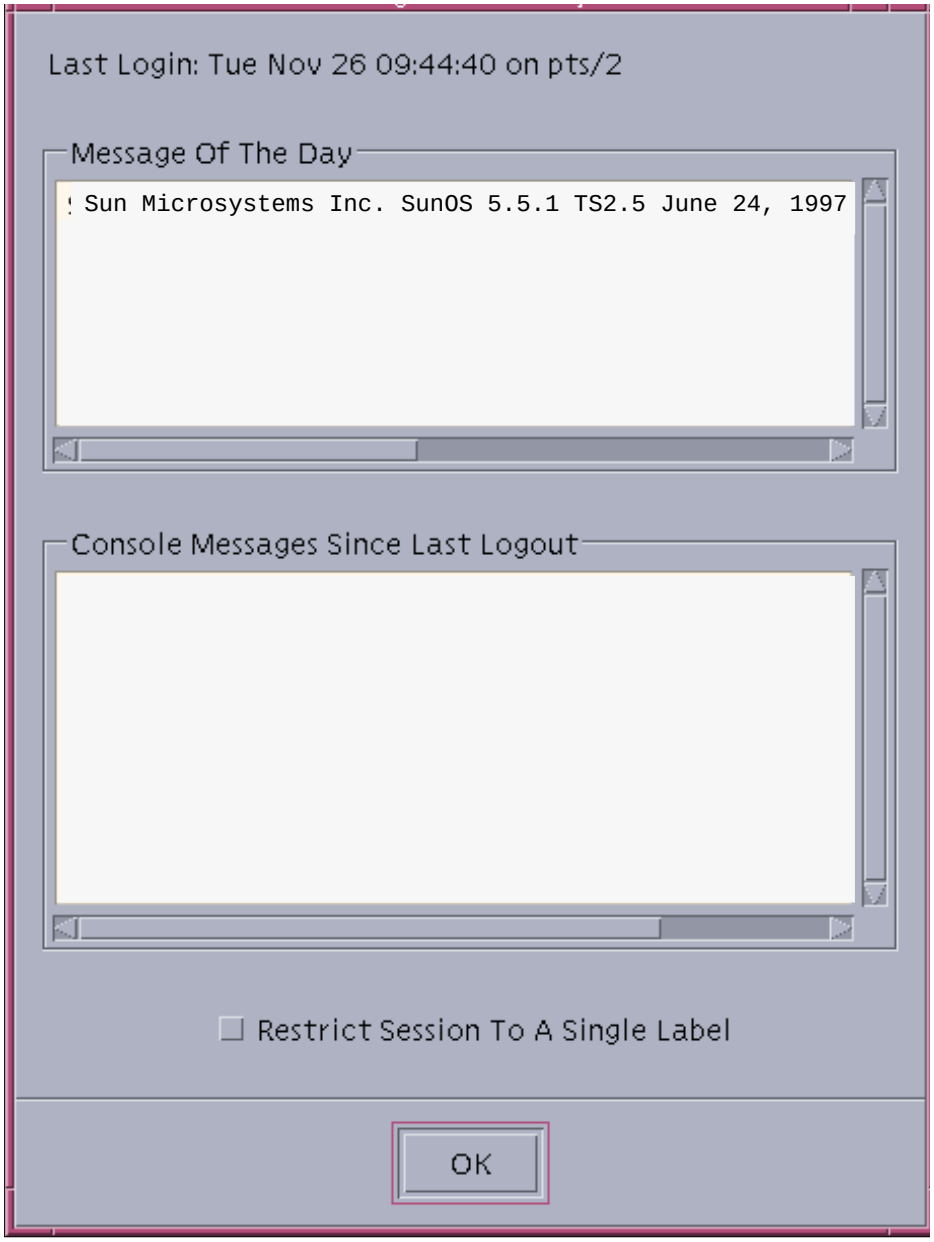



*Figure 1-8* Workstation Information Dialog Box

3. **Review the information provided on the workstation information dialog box (Figure 1-8 on page 14) and, if allowed, choose between a single sensitivity label or multiple sensitivity label session.**

   a. **Check the date and time of the last login to ensure there is nothing suspicious about the last login, such as an unusual time of day.**

   b. **Read the message of the day.**

   c. **Check console messages since last logout.**

   d. **Report any suspicious logins, messages that could indicate suspicious activity, or other problems to your the administrator or system administrator, as appropriate.**

   e. **If your user account is configured to be able to work with multiple labels, check or ignore the Restrict Session to a Single Label toggle box, depending on whether you want to start work in a multilabel or single-label session.**

---

**Note –** If your account is configured to work at only one sensitivity label, “Single Level Session SL:” displays at the bottom of the workstation information dialog box followed by the sensitivity label at which you are configured to work. For example, if your single sensitivity label is CONFIDENTIAL, the text shown in Figure 1-9 displays.

---

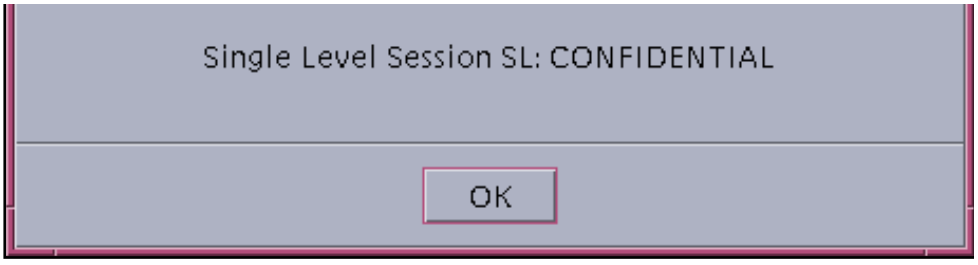



*Figure 1-9* Single Label Indicator on the Message of the Day Dialog Box

4. **Press Return or click OK to close the workstation information dialog box.**

   If you checked the box next to Restrict Session to a Single Label, the Setting User Session SL dialog box displays (see Figure 1-10 on page 17 and go to Step 5 on page 16).

If you are allowed to work at multiple sensitivity labels and if you did not check the box next to Restrict Session to a Single Label, on the Workstation Information dialog box, the Setting User Session Clearance dialog box displays (see Figure 1-11 on page 19 and go to Step 6 on page 18).

5. **To specify a sensitivity label for a single-label session, either accept the default sensitivity label or specify a sensitivity label in the Single Level Session: Setting User Session SL Label Builder.**

   a. **To type in the sensitivity label, use the text entry field under Update With, and click Update when done.**

   b. **To use the mouse to build the sensitivity label, choose a classification from the CLASS menu and select the compartments by checking the boxes next to the compartment names under COMPS.**

   c. **Click OK.**

   d. **Go to Step 7 on page 19.**

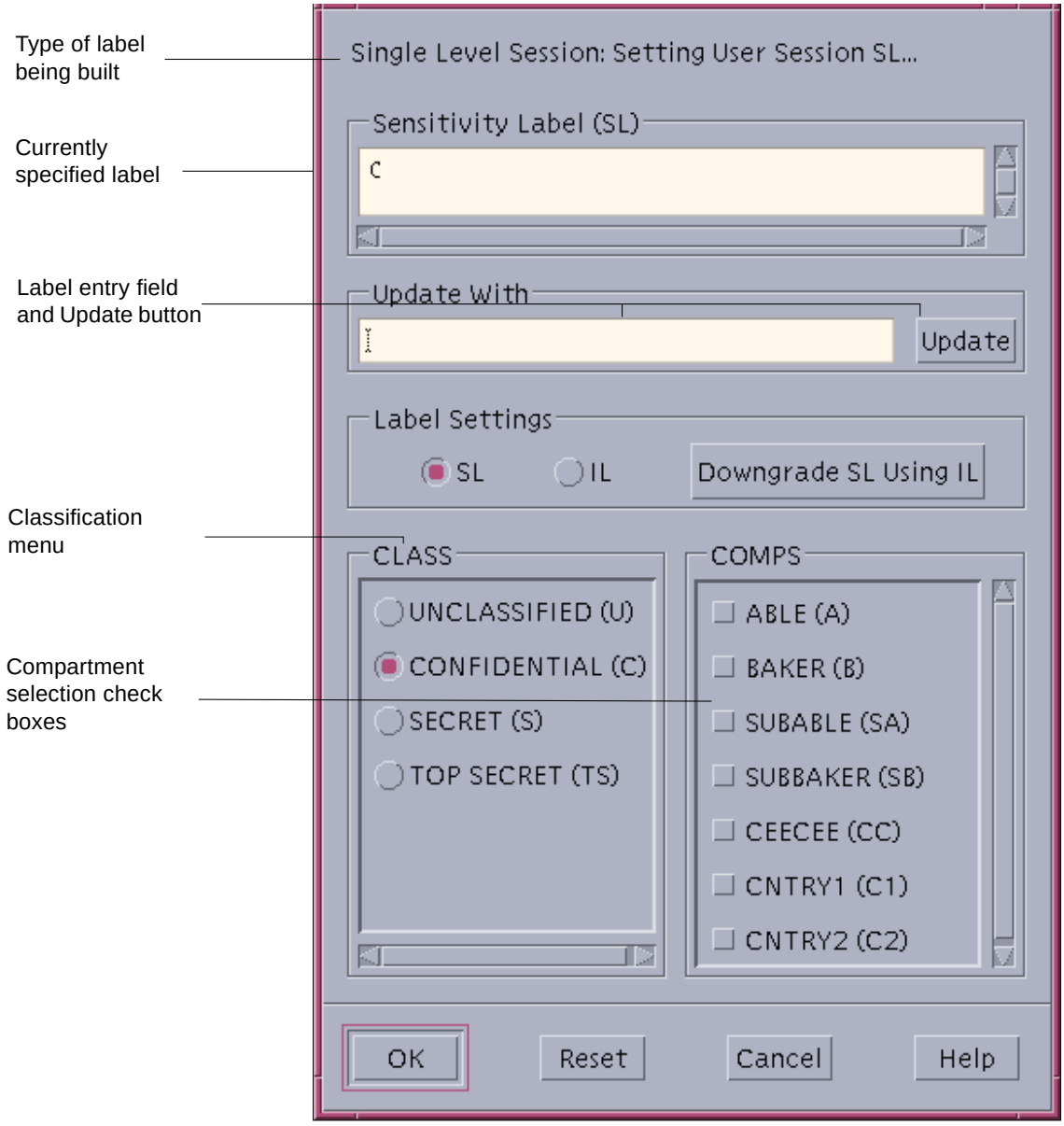



*Figure 1-10* Label Builder Dialog Box for a Single-sensitivity Label Session

6. **To specify the clearance for a multilabel session, either accept the default clearance shown in the Multi Level Login Setting User Session Clearance label builder, or specify another clearance.**

   a. **To type in the clearance, use the text entry field below Update With and click Update when done.**

   b. **To use the mouse to build the clearance, choose a classification from the Class menu and select the compartment components by checking the boxes next to the compartment names in the Comps column.**

   c. **Click OK.**

   d. **Go to Step 7 on page 19.**

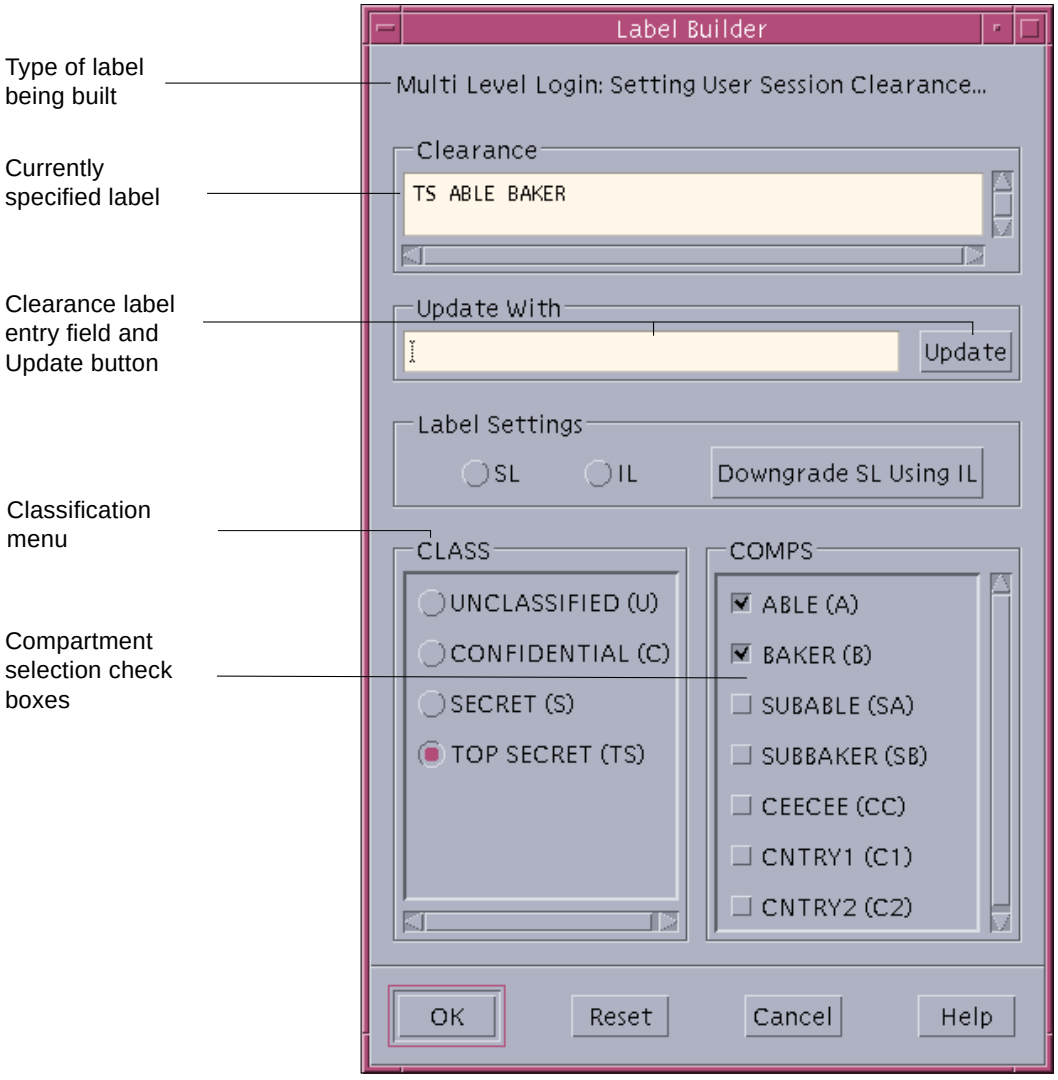



*Figure 1-11* Session Clearance Dialog Box

7. **Choose the assume role option from the Trusted Path menu**
   Figure 1-12 on page 20 shows the Trusted Path menu for a user who is configured to assume the system administrator role.

**Note –** The option Assume *role_account_name* role appears on the Trusted Path menu only for users who are configured to assume a role. A user configured for the security administrator role sees `Assume secadmin Role`; a user configured for the system administrator role sees `Assume admin Role`, and so on. (secadmin is the account name for the security administrator role and admin is the account name for the system administrator role).

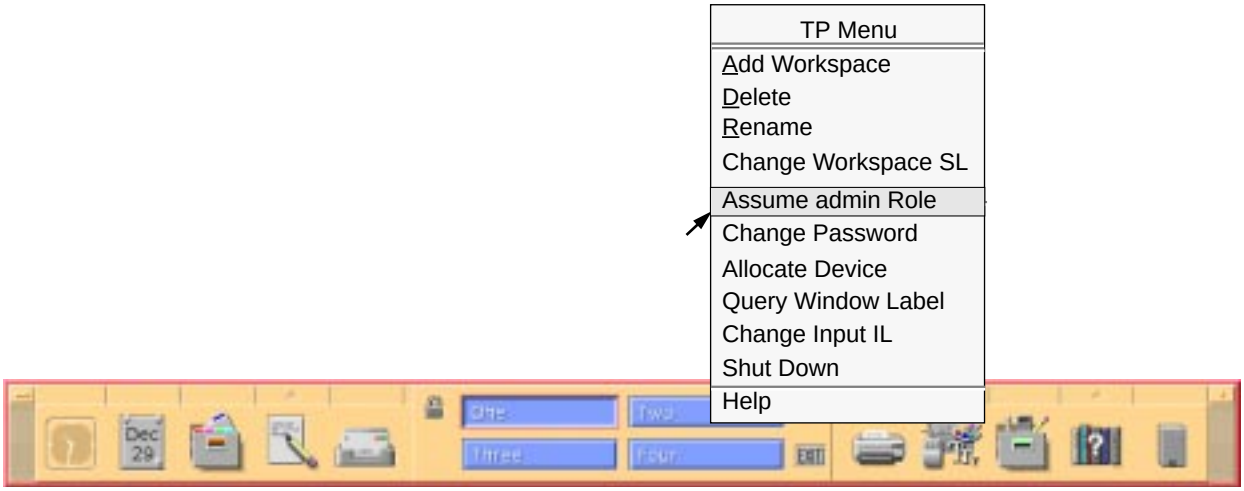



*Figure 1-12* Choosing the Assume admin Role Option from the Trusted Path Menu

The Role Password dialog box displays.

8. **Enter the role password in the role password dialog box (see Figure 1-13), and click OK.**

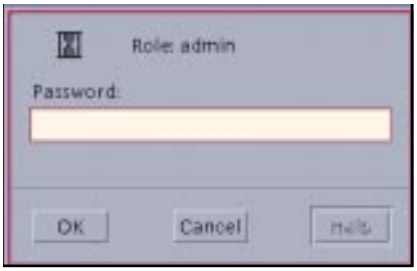



*Figure 1-13* Role Password Dialog Box

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

### ▼ To Switch Among Administrative Role Workspaces and the Normal User Workspaces

♦ **Click the desired workspace's button.**

### ▼ To Work at Multiple Labels While in an Administrative Role

1. **Add a new administrative role workspace.**

   a. **Move the cursor to the button for a role workspace.**

---

**Note –** If you do not have the cursor over an administrative role workspace's button, but instead have it somewhere else in the workspace switch area, Add Workspace adds a normal user workspace.

---

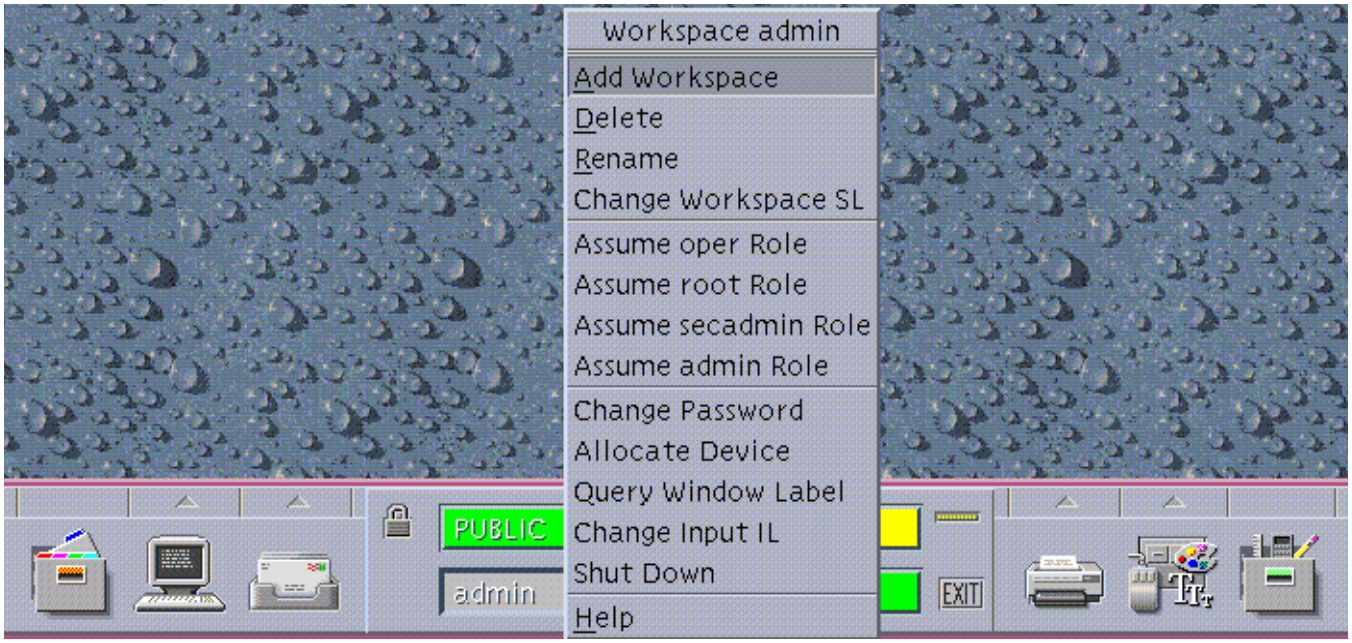



*Figure 1-14* Creating a New Role Workspace From the Trusted Path Menu

   b. **Click the right mouse button on the workspace switch button to bring up the Trusted Path menu (see Figure 1-14).**

**c. Choose Add Workspace**

A new administrative role workspace becomes active, and a new administrative role workspace button appears in the workspace switch area in the Front Panel, as shown in Figure 1-15.

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`.

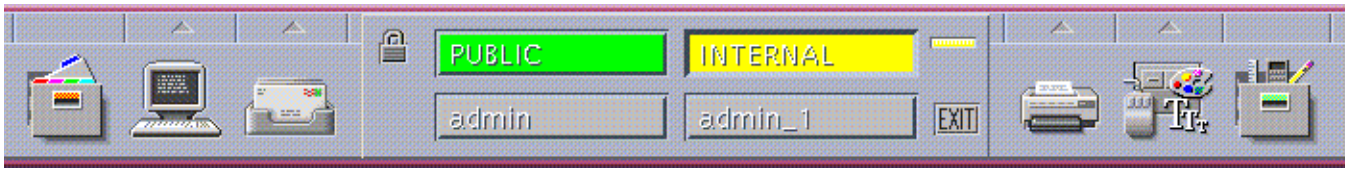



*Figure 1-15* A New admin_1 Workspace Button for a New Administrative Role Workspace

**2. Change the sensitivity label of the workspace.**

**a. Move the cursor to the new role workspace button.**

**b. Click the right (MENU) mouse button on the workspace switch button to bring up the Trusted Path menu.**

**c. Choose Change Workspace SL from the Trusted Path menu.**
See Figure 1-16.

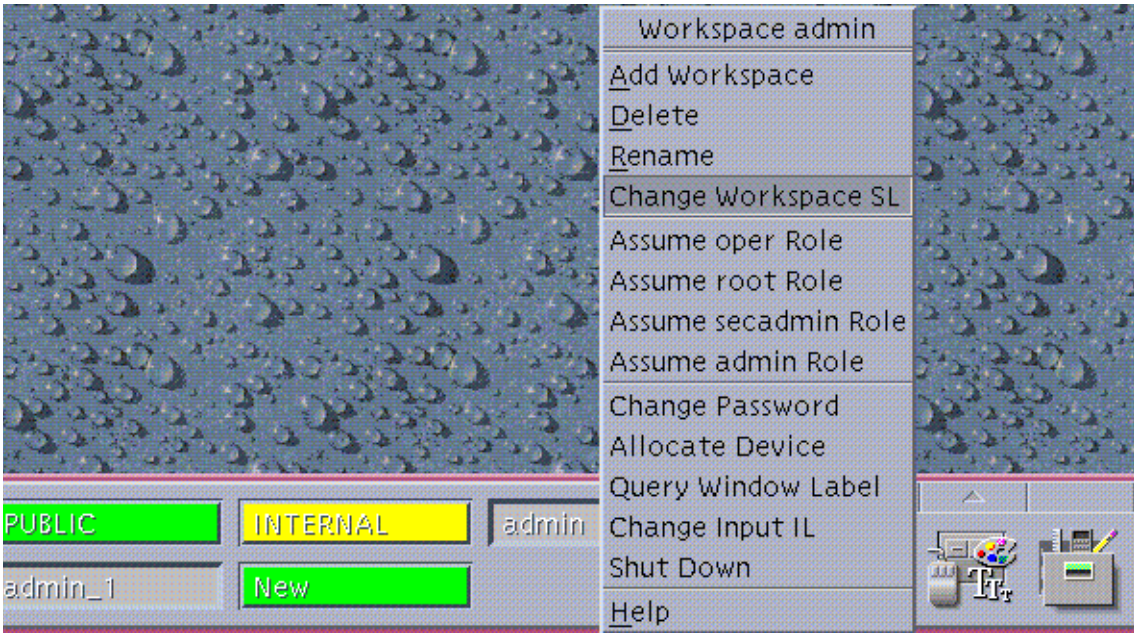



*Figure 1-16* Relabeling an Administrative Role Workspace

The Label Builder displays.

**d. Type in the desired sensitivity label in the text entry field under Update With on the Label Builder dialog box, and then click the Update button and click OK at the bottom of the dialog box when done.**
The sensitivity label on the workspace changes to the sensitivity label you specified in the label builder.

## ▼ To Launch Administrative Applications

1. **Click the Applications Manager icon in the Front Panel to open it.**
2. **Double-click the Solstice_Apps folder.**
3. **Double-click the icon for the desired application.**

### ▼ To Launch Administrative Actions

1. **Click the Applications Manager icon in the Front Panel to open it.**
2. **Double-click the System_Admin folder**
3. **Double-click the icon for the desired action.**

### ▼ To Create A New Administrative Action for Editing an Administrative File

1. **As security administrator in an ADMIN_LOW workspace, use the Admin Edit action to open the** `/usr/dt/appconfig/types/C/TSOLadmin.dt` **file.**
2. **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
}
```

3. **Modify the copied action's definitions.**
   a. **Change the ACTION name.**
      This example creates a new action to edit the `system`(4) file to modify Trusted Solaris kernel switch settings.

```
ACTION EditSwitches
{
```

   b. **Change the LABEL.**

```
        LABEL           Set TSOL Switches
```

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 TSOL-related kernel switches
}
```

4. **Save and Close the** `TSOLadmin.dt` **tile.**

5. **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 `EditSwitches`.

   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 EditSwitches
```

6. **To make the action available to an administrative role on all hosts in the distributed system, copy the modified** `TSOLadmin.dt` **and action files to the NIS+ master and to all hosts in the distributed system, and bring up the Profile Manager, choosing NIS+ as the naming service.**
   Since the actions are not administered through NIS+, some other means of distribution must be used, such as `rdist(1)` or sneakernet (copying the files to a tape or floppy and carrying it around to install the files on each host).

7. **To make the action available only on one host, bring up the Profile Manager, choosing None as the naming service.**
8. **Choose a profile, either the System Security profile or the System Management profile and choose Actions from the View menu.**
   The new action should be listed in the System_Admin application group in the Excluded list of actions.
   a. **If the action edits a security-relevant file [such as the `system`(4) file] assign the action to the System 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`(4) file] assign the action to the System Management profile.**
9. **Assign to the new 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.**
10. **Log out and log in again.**

### ▼ To Add Actions Outside 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 the limits of the Trusted Solaris MAC restrictions. Actions can be created either by using CreateAction or manually. The action should be placed in the `/etc/dt/appconfig/types/C` directory.

### ▼ To Prevent Logins From Being Disabled After a Reboot

1. **Assume the security administrator role.**
2. **In an ADMIN_LOW workspace, use the file manager or commands in a terminal emulator to make a backup copy of `/etc/rc2.d/S05RMTMPFILES`.**

```
$ cd /etc/rc2.d
$ cp S05RMTMPFILES S05RMTMPFILES.orig
```

3. **Use the Admin Edit action to open the** S05RMTMPFILES for editing.

4. **Find the lines shown in the first screen below and comment out the active lines as shown in the second screen.**

```
# The file /etc/nologin will disable all logins until the
# logins are enabled by an authorized user via login(1)
# or dtlogin(1).

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
```

```
# 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
```

5. **Save and close the file.**

# 1

# Miscellaneous Tasks and Procedures 2

This chapter provides information about tasks that do not fit neatly under the divisions of tasks in other chapters in this manual and procedures for performing those tasks. It includes the following main topics:


| | |
|---|---|
| *Getting a Hexadecimal Equivalent for Labels and Clearances* | *page 30* |
| *Extending Extendable Security Mechanisms* | *page 31* |
| *Understanding Authorizations* | *page 32* |
| *Extending the Trusted Solaris Authorizations* | *page 33* |
| *Extending the Trusted Solaris Privileges* | *page 37* |
| *Changing the Trusted Stripe* | *page 42* |
| *Working with MLDs* | *page 43* |
| *MLD Prefix/MLD Adornment* | *page 44* |
| *How SLDs Are Created* | *page 44* |
| *How SLDs Are Named* | *page 44* |
| *Restriction on the Creation of MLDs and Its Effects* | *page 45* |
| *MLD and SLD Prefixes* | *page 46* |
| *Creating, Changing, Finding Your Way Around In, and Deleting MLDs* | *page 46* |

# 2

This chapter includes the following procedures:


| | |
|---|---|
| *To Add An Authorization* | *page 36* |
| *To Add a Privilege* | *page 40* |
| *To Change the Font or Color of the Trusted Path Indicator* | *page 42* |
| *To Find Out if a Directory is an MLD* | *page 49* |
| *To Create an MLD from the File Manager* | *page 49* |
| *To Create an MLD from the Command Line* | *page 49* |
| *To Identify an MLD* | *page 50* |
| *To Identify an SLD* | *page 51* |
| *To Address the Entire MLD* | *page 51* |
| *To Change the MLD Prefix of a File System* | *page 51* |
| *To Remove an MLD* | *page 52* |


## *Getting a Hexadecimal Equivalent for Labels and Clearances*

Administrators need to enter labels and clearances into the trusted network databases and other configuration files in hexadecimal form. Use `atohexlabel(1MTSOL)` with the options shown in the following steps to get the hexadecimal equivalents of a CMW label, sensitivity label, information label or clearance.

### ▼ To Get a Hexadecimal Equivalent for a CMW Label, an SL, IL, or Clearance

1. **Assume the security administrator role, go to an ADMIN_LOW workspace, bring up a profile shell [`pfsh(1MTSOL)`].**
2. **To get the hexadecimal value for an ASCII CMW label (an information label followed by a sensitivity label in brackets), use `atohexlabel(1MTSOL)` followed by the ASCII name of the CMW label.**

```
$ /usr/sbin/atohexlabel ADMIN_LOW
0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000[0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000]
```

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

```
$ atohexlabel -s s
    0x00060c0000000000000000000000000000000000000000000003fffffffffff0000
```

4. **To get the hexadecimal value for an ASCII information label, use `atohexlabel (1MTSOL)` with the -s option followed by the ASCII name of the sensitivity label.**

```
$ atohexlabel -i s
    0x00060c0000000000000000000000000000000000000000000003fffffffffff0000
```

5. **To get the hexadecimal value for an ASCII clearance name, use `atohexlabel(1MTSOL)` with the `-c` option followed by the ASCII name of the clearance.**

```
$ atohexlabel -c sa
    0x0006ac0000000000000000000000000000000000000000000003fffffffffff0000
```

## *Extending Extendable Security Mechanisms*

The following Trusted Solaris security mechanisms can be added to by the security administrator role:

- Audit events
- Audit classes
- Execution profiles
- Roles
- Authorizations
- Privileges

Adding audit events and audit classes is described in the *Trusted Solaris Audit Administration*. Adding execution profiles is described in the current manual in Chapter 8, “Managing Execution Profiles for Users and Roles” under “Use of the Profile Manager to Create or Modify Execution Profiles” on page 198.

Adding roles is described in Chapter 4, “Managing Roles” under “Creating a New Role” on page 95. The rest of this section describes how to add authorizations and privileges.

## *Understanding Authorizations*

Authorizations are granted to users only indirectly through the execution profiles that may be assigned to the user.

Zero or more authorizations are assigned to each execution profile through the Solstice Profile Manager, and are stored in the authorization field in the `tsolprof(4TSOL)` entry for each profile. The authorizations field contains either a comma-separated list of authorization numbers or names or the key words *all* or *none*. Figure 2-1 shows the format of a `tsolprof` entry.

```
profile:description:authorizations:actions:commands:links:flags;
```

*Figure 2-1* Format of a `tsolprof` Entry

As shown in Figure 2-2 the Audit Review profile has the terminal login authorization (#3) and the set file privilege authorization (#9).

```
Audit Review:View The Audit Trail:none:none:/var/sbin;praudit;3,9;none;none,0x00000000000
0000000000000000000000000000000000000000000000000000000,0x7fffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffff:1:none
```

*Figure 2-2* An Example `tsolprof` Entry

When setting up or modifying an account, the security administrator role assigns zero or more execution profiles using the Solstice User Manager. The names of the assigned profiles are stored in the profiles field in the `tsoluser(4TSOL)` entry for that account.

Figure 2-3 shows the comma-separated list of profile names in the profile field of the `tsoluser` entry for a locally-created administrative role account, *auditadmin*.

```
auditadmin:fixed:automatic:Audit Control,Audit Review,Media Restore,:none:5:lock:internal,showil,showsl:
0x0000:0x00000000000000000000000000000000000000000000000000000000000000000000000000000000:0x7fffffff
```

TSOL_AUTH_* authorizations are already defined; the TSOL_AUTH definitions for the default authorizations number from 1 to 27 (with 0 meaning no authorizations).

```
    TSOL_AUTH_ENABLE_LOGIN = 1,/* comment */
    TSOL_AUTH_REMOTE_LOGIN = 2,/* comment */
    TSOL_AUTH_TERMINAL_LOGIN = 3,/* comment */
                        .
                        .
                        .
    TSOL_AUTH_USER_HOME = 25,/* comment */
    TSOL_AUTH_PRINT_POSTSCRIPT = 26,/* comment */
    TSOL_AUTH_PRINT_UNLABELED = 27,/* comment */
```

*Figure 2-4* TSOL_AUTH Defined Authorizations in auth_names.h

As shown in Figure 2-5, an additional thirty-six (36) authorizations (from 28 up to 63) are reserved for later Sun extension, and these are identified by the name TSOL_AUTH_RESERVED.

```
    TSOL_AUTH_RESERVED_028 = 28,/* comment */
    TSOL_AUTH_RESERVED_029 = 29,/* comment */
    TSOL_AUTH_RESERVED_030 = 30,/* comment */
                        .
                        .
                        .
    TSOL_AUTH_RESERVED_062 = 62,/* comment */
    TSOL_AUTH_RESERVED_063 = 63,/* comment */
```

*Figure 2-5* TSOL_AUTH_RESERVED Authorizations in auth_names.h

As shown in Figure 2-6, the remaining sixty-four (64) authorizations identified with the AUTH_RESERVED name are available for any site to extend.

```
AUTH_RESERVED_064 = 64,/* comment */
AUTH_RESERVED_065 = 65,/* comment */
AUTH_RESERVED_066 = 66,/* comment */
AUTH_RESERVED_067 = 67,/* comment */
                    .
                    .
                    .
AUTH_RESERVED_125 = 125,/* comment */
AUTH_RESERVED_126 = 126,/* comment */
AUTH_RESERVED_127 = 127/* comment */
```

*Figure 2-6* Authorizations Available for Extension

## auth_name(4TSOL)

Figure 2-7 shows the format for an entry in auth_name(4TSOL).

*constant*:*name*:*description*

*Figure 2-7* Format of the auth_name File

The constant value must be exactly the same as the manifest constant name for the authorization in the /usr/include/tsol/auth.h file. The name field is a concise and descriptive name for the authorization, which is used in various GUIs, including the profile manager.

**Note –** The authorization name (in the example in Figure 2-8, enable_logins) is displayed in the Profile Manager and other GUIs that display a list of authorizations.

The description field is a description of the activity permitted by the authorization. The definition may be as long as you need it to be. The definition is used in the Profile Manager to guide the security administrator when assigning authorizations to profiles.

Figure 2-8 gives an example of an actual authorization in the default `auth_name` file. Note that the manifest constant name is in all capital letters, the name is in lowercase letters, and the description is continued from line to line with the backslash character (\).

```
TSOLAUTH_ENABLE_LOGIN:enable logins:Allows a user to enable logins on a \

    machine that was just booted.  Until logins are enabled there is \

    no interactive use of the machine's resources.
```

*Figure 2-8* Definition for the enable logins Authorizations in the `auth_name` File

## ▼ To Add An Authorization

1. **Assume the security administrator role and use the Admin Edit action to open the `auth_names.h` file for editing.**
2. **Contact your Trusted Solaris representative to reserve an authorization number.**
3. **Create an entry in the `auth_names.h` file with the manifest constant for the privilege (see Figure 2-9).**

```
AUTH_POWER = 127,/* To leap tall MAC check failures and otherwise be irresponsible and
       unaccountable*/
```

*Figure 2-9* Specifying a Manifest Constant for a New Authorization in `auth_names.h`

4. **Save and close the file.**
5. **As security administrator, use the Admin Edit action to open the `/usr/lib/tsol/locale/C/auth_name` file for editing.**
6. **Create an entry with the manifest constant, name, and definition for the authorization in the `auth_name` file (see Figure 2-10).**

**Caution –** Make sure that the first field has the identical manifest constant defined for the authorization in the `auth_names.h` file in Figure 2-9.

```
AUTH_POWER:power user:Allows a user to bypass all MAC and DAC
checks and auditing flag settings and to be otherwise totally
unaccountable for his or her actions.
```

*Figure 2-10* Specifying a Name and a Description For A New Authorization in `auth_name`

**7. Save and close the file.**

## *Extending the Trusted Solaris Privileges*

The use of privileges is discussed in great detail elsewhere in the Trusted Solaris document set, so this section focuses on how to add 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`

### `priv_names.h`

The `/usr/include/sys/tsol/priv_names.h` header file contains the manifest constants and associated numbers for privileges. Up to 128 possible privileges are allowed. As shown in Figure 2-11, approximately eighty-six (86)

`privileges` are already defined; the definitions for the default privileges number from 1 to 86(with 0 meaning no privileges, and the number 62 retired, as is explained below).

```
    PRIV_FILE_AUDIT = 1,/* operational */
    PRIV_FILE_CHOWN = 2,/* operational */
    PRIV_FILE_DAC_EXECUTE = 3,/* policy */
                        .
                        .
                        .
    PRIV_WIN_SELECTION = 84,/* operational */
    PRIV_WIN_UPGRADE_IL = 85,/* operational */
    PRIV_WIN_UPGRADE_SL = 86,/* operational */
```

*Figure 2-11* Manifest Constants and Numbers for Default Privileges in `priv_names.h`

As shown in Figure 2-12, four (4) `privileges` are reserved for Trusted Solaris extension, and these are identified by the name `tsol_reserved`. The number 62 at the top of the reserved list is the number of a retired privilege, which is a privilege that once was defined but now is not. The policy, which is explained at the top of the file, is for anyone retiring a privilege to add it to the top of the reserved list, and for any Sun Federal developer adding a new Trusted Solaris privilege to get the number from the top of the reserved list.

```
/* Reserved for Trusted Solaris */

    tsol_reserved62 = 62,
    tsol_reserved87 = 87,
    tsol_reserved88 = 88,
    tsol_reserved89 = 89,
```

*Figure 2-12* Privilege Numbers Reserved for Trusted Solaris Use

As shown in Figure 2-13, the remaining privileges identified with the reserved name are available for any site to extend.

```
/* Reserved for ISV, GOTS, integrator, ... use */

    reserved90 = 90,
    reserved91 = 91,
    reserved92 = 92,
                        .
                        .
                        .
    reserved126 = 126,
    reserved127 = 127,
    reserved128 = 128
```

*Figure 2-13* Privileges Available for Extension

## priv_names(4TSOL)

Figure 2-14 shows the format for an entry in `priv_names(4TSOL)`.

*constant*:*name*:*description*

*Figure 2-14* Format of the `priv_name` File

The constant field must be exactly the same as the manifest constant name for the privilege in the `/usr/include/sys/tsol/auth.h` file. The name field is the name of the privilege, which must be concise and descriptive for use in various GUIs.

**Note –** The privilege name (in the example in Figure 2-15, file_audit) is displayed in the Profile Manager, File Manager, and other GUIs that display a list of privileges.

The description field is a description of the activity permitted by the privilege. The definition may be as long as you need it to be. The definition is used in the Profile Manager to guide the security administrator when assigning privileges to programs.

Figure 2-15 gives an example of an actual privilege in the default `priv_name` file. Note that the manifest constant name is in all capital letters, the name is in lowercase letters, and the description is continued from line to line with the backslash character (\).

```
PRIV_FILE_AUDIT:file_audit:Allows a process to get or set a file's or \
       directory's audit preselection information. The auditing preselection \
       information may override the preselection information associated with \
        a process' access to a file or directory. \
        Allows a process to get or set a file's or directory's public object \
        flag. The public object flag may override the successful read/search \
        access preselection information associated with a process' access to \
        a file or directory.
```

*Figure 2-15* Definition for the file_audit privilege in the `priv_name` File

### ▼ To Add a Privilege

1. **Assume the security administrator role and use the Admin Edit action to open the `/usr/include/sys/tsol/priv_names.h` file for editing**

**2. Read the comment at the top of the `priv_names.h` file (see Figure 2-16).**

```
/ *
 * ********************** 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', integrator's 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(3TSOL) 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.
```

*Figure 2-16* Comment from the `priv_names.h` File

**3. Contact your Trusted Solaris representative to reserve a privilege number.**

4. **Create an entry in the** `priv_names.h` **file with the manifest constant for the privilege (see Figure 2-17).**

```
PRIV_RISK = 90,/
```

*Figure 2-17* Specifying a Manifest Constant for a New Privilege in `priv_names.h`

5. **Save and close the file.**

6. **As the security administrator, use the Admin Edit action to open the** `/usr/lib/tsol/locale/C/priv_names` **file for editing.**

7. **Create an entry with the manifest constant, name, and definition for the privilege in the** `priv_name` **file (see Figure 2-18).**

**Note –** Make sure that the first field has the identical manifest constant defined for the privilege in the `priv_names.h` file in Figure 2-17.

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

*Figure 2-18* Specifying a Name and a Description for a New Privilege in `priv_name`

8. **Save and close the file.**

## *Changing the Trusted Stripe*

The trusted path indicator (TP) in the trusted stripe displays in inverted colors (swapped background and foreground) to make it easier to see with your peripheral vision. It is also now displayed in bold font.

### ▼ To Change the Font or Color of the Trusted Path Indicator

1. **Assume the security administrator role and in an ADMIN_LOW workspace, use the Admin Edit action to open the** `/usr/dt/app-defaults/C/Dtwm` **file for editing.**

2. **Change the Dtwm*ss_key_icon definition as desired.**
   The lines in the following example make the text display in bold.

```
!#ifdef TSOL

Dtwm*ss_key_icon*fontList: \

    -dt-*-bold-r-normal-s*-14-*-*-*-*-*-*-*:bold

!#endif TSOL
```

Adding the line in the following example sets the foreground color to Red.

```
Dtwm*ss_key_icon*foreground:Red
```

## *Working with MLDs*

Multilevel directories are called MLDs, and single-level directories are called SLDs. As described in the *Trusted Solaris User's Guide*, files and directories at differing sensitivity labels in an MLD are automatically and transparently segregated within SLDs at differing sensitivity labels, while the existence of the SLDs is hidden during ordinary use.

MLDs were created to allow applications that are running at differing sensitivity labels to write into what appears to be the same directory. `/tmp` is the directory most often used by multiple applications, and for that reason, `/tmp` is an MLD in the default system. Applications are not aware that when they write a file into `/tmp`, they are actually writing the file into an SLD within `/tmp` that has the sensitivity label at which the application is running. Home directory MLDs allow accounts to create files and folders at different sensitivity labels within their home directories. When user or role accounts change into their home directories, they do not need to be aware that they have actually changed into an SLD that is at the same sensitivity label as their current workspace.

For example, when setting up a new account for user roseanne, the User Manager creates the home directory `/export/home/roseanne` as an MLD. When the user changes to her home directory, for example by entering `cd ~` or `cd /export/home/roseanne` on the command line, she is automatically and

transparently redirected to an SLD within her home directory MLD. The SLD has the same sensitivity label as her current process, so if she changes to her home directory while in a NEED_TO_KNOW workspace, she actually changes into the SLD that has the NEED_TO_KNOW sensitivity label.

The rest of this section covers the prerequisite information and describes how to work with MLDs and how to refer to an MLD as a whole so you can see and directly access any SLDs that it contains.

## *MLD Prefix/MLD Adornment*

The MLD prefix is also called the *MLD adornment*, and, when the name of an MLD includes the *MLD prefix*, it is referred to as the *adorned name*. The adorned name of an MLD is used to directly access the top level directory where the SLDs are stored. The MLD prefix may be changed by a site as described in “To Change the MLD Prefix of a File System” on page 51.

In the default system, the MLD prefix is ”`.MLD.`”. For example, the adorned name for the `/tmp` MLD is `/.MLD.tmp`.

If an MLD is accessed using its *unadorned* name, the application goes to the appropriate SLD at the same sensitivity label within the MLD, while if an MLD is accessed by its *adorned* name, the application goes to the top level of the MLD and can view individually all the SLDs the MLD contains.

## *How SLDs Are Created*

Each time a user accesses an MLD at a new sensitivity label for the first time, a new SLD at the sensitivity label of the current process is created

The sensitivity label of the first SLD created for a new user at first login is at the user’s minimum sensitivity label. User roseanne’s minimum sensitivity label is PUBLIC, so the sensitivity label of the first SLD created for her when she logs in for the first time is PUBLIC.

## *How SLDs Are Named*

SLDs always have the prefix `.SLD.`, and, unlike the MLD prefix, the SLD prefix cannot be changed. SLD names are given sequential numbers as new SLDs are created within an MLD. The name of the first SLD created in an MLD is numbered 0 (.SLD.0), the second is numbered 1 (.SLD.1), and so forth.

When working in a PUBLIC workspace, if roseanne uses `pwd` on the command line or views her home directory folder in the File Manager, she sees the name of the directory as `/export/home/roseanne`, but while she apparently is working in her home directory, she is actually in an SLD whose sensitivity label is PUBLIC and whose adorned name is `/export/home/.MLD.roseanne/.SLD.0`.

The numbers in the SLD names have no relation to the sensitivity label of the directory. For example, `/export/home/.MLD.roseanne/.SLD.0` is given the number 0 because it is the first SLD created for the user at any sensitivity label

The MLD keeps track of the label of each SLD, and each time the MLD is accessed a second and subsequent time at a particular sensitivity label, the user is transparently changed into the SLD at the correct sensitivity label.

With the first SLD created at PUBLIC and assigned the name `.SLD.0`, the next time roseanne changes to her home directory while running at PUBLIC, she actually changes transparently into the PUBLIC SLD, `.SLD.0`, and any files she creates are stored within `.SLD.0`. All this is transparent to the user unless the user directly refers to the MLD or SLD using the prefixes, as described under “MLD and SLD Prefixes” on page 46.

## *Restriction on the Creation of MLDs and Its Effects*

An MLD can only be created within a directory that is not an MLD.

The effect of this restriction is that normal users cannot create MLDs in the default system because they cannot write into any directory other than their home directories and because their home directories are MLDs and they cannot create an MLD within an MLD  If the system administrator creates a new directory that is not an MLD and that is writable by normal users, only then will it be possible for normal users to create MLDs.

For example, the system administrator at one site has created a directory called `/shark/doc` mounted by and writable by all developers at a single sensitivity label, so that design specifications and other project-wide documentation can be kept in one commonly-accessible place. Anyone in the development group can create new directories within that directory, and anyone can change an existing directory to an MLD. See “Creating, Changing, Finding Your Way Around In, and Deleting MLDs” on page 46 for more about this topic.

## MLD and SLD Prefixes

The adorned name of the MLD is used to access the top level directory in an MLD where the SLDs at various sensitivity labels are stored.

Using the adorned name of an MLD allows you to bypass the transparent redirection to an SLD within an MLD, lets you go to the MLD directly, and lets you view each of the single-level directories it contains. Similarly, using the adorned name of an SLD brings you directly to the SLD itself.

The following screen shows a listing of the `/tmp` MLD using the `.MLD.` prefix at the NEED TO KNOW ENG label, and shows the labels of the SLDs. `.SLD.1` and `.SLD.2` are at sensitivity labels that dominate the sensitivity label at which `getlabel` was invoked, so the error `Not owner` displays for them.

```
trustworthy% ls /.MLD.tmp/.*

/.MLD.tmp/.SLD.0
bt+_errlog.26629      dtbcache_:0
bt+_errlog.26630      ps_data

/.MLD.tmp/.SLD.4
bt+errlog.631         mpa000_M
mailBAAa000K4         ps_data
mpa000.E              sel_mgr.err

trustworthy% getlabel /.MLD.tmp/.*
.SLD.0 PUBLIC [PUBLIC]
.SLD.1 Not owner
.SLD.2 Not owner
.SLD.4 PUBLIC [NEED TO KNOW ENG]
```

## Creating, Changing, Finding Your Way Around In, and Deleting MLDs

Anyone who can write into a directory (who has the DAC and MAC write access or has the authorization to bypass MAC when writing into a directory or has the commands required with the needed privileges) can create a new MLD or change an existing directory into an MLD.

- Create a new MLD
  - Using the File Manager as shown in “To Create an MLD from the File Manager” on page 49

- Using the mkdir command as shown in "To Create an MLD from the Command Line" on page 49

- Change any directory to an MLD as shown in "To Create an MLD from the Command Line" on page 49

Other useful commands and options for working with MLDs are listed in Table 2-1. See the man pages for complete information.

*Table 2-1* MLD-related Commands and What They Do *(1 of 2)*

| MLD-related Commands and Options | What They Do |
|---|---|
| **`adornfc`** *dirname*<br>See `adornfc(1TSOL)` | Display the pathname of a directory with the final component adorned (with the MLD prefix that is set for the filesystem, which is .MLD. by default). Does exactly as described, displaying the pathname with the MLD prefix, whether or not the final component is an MLD and whether or not it is a directory. This command does not *change* anything. |
| **`getfattrflag -m`** *dirname*<br>See `getfattrflag(1TSOL)` | Tells you whether or not the directory is an MLD. |
| **`getmldadorn`** *filesystem_name*<br>See `getmldadorn(1TSOL)` | Displays the MLD prefix of the file system. |
| **`getsldname`** *filesystem_name*<br><br>OR<br>**`getsldname`** `-s` *sensitivity_label filesystem_name*<br><br>See `getsldname(1TSOL)` | Displays the single-level directory name for the file system.<br>Displays the SLD name that corresponds to the sensitivity label of the current process.<br><br>Displays the single-level directory name for the SLD that has the specified sensitivity label file system. |
| **`mldpwd`**<br>See `mldpwd(1TSOL)` | Displays the pathname of the working directory including the MLD prefix if it is an MLD and the SLD name, if it is an SLD. |
| **`mldrealpath`** *dirname*<br>See `mldrealpath(1TSOL)` | Displays the pathname of the specified directory including the MLD prefix if it is an MLD and the SLD name, if it is an SLD. |

*Table 2-1* MLD-related Commands and What They Do *(2 of 2)*

| MLD-related Commands and Options | What They Do |
|---|---|
| **rm -RM** *MLD_dirname*<br><br>See rmdir(1TSOL) | Recursively removes all SLDs to which it has DAC and MAC access within the specified MLD |
| **mkdir -M** *dirname*<br>OR<br>**mkdir** *.mld_prefix.dirname*<br>See mkdir(1TSOL) | Make a new MLD. |
| **setfattrflag -m** existing.dirname<br>See setfattrflag(1TSOL) | Change an existing directory to an MLD |

To illustrate the relationship between the MLD and the SLD and the sensitivity label of the process, Figure 2-19 shows the output of the pwd, plabel, and mldpwd commands in a terminal at the CLASSIFIED sensitivity label, when the user's home directory (/export/home/roseanne) is an MLD, and .SLD.2 is the SLD created at the CLASSIFIED sensitivity label. In Figure 2-19, the normal user command pwd shows the home directory name, while mldpwd shows that the user is actually in .SLD.2 within the .MLD.roseanne directory. The CLASSIFIED SLD was created third, so it has an SLD number of 2: .SLD.2.

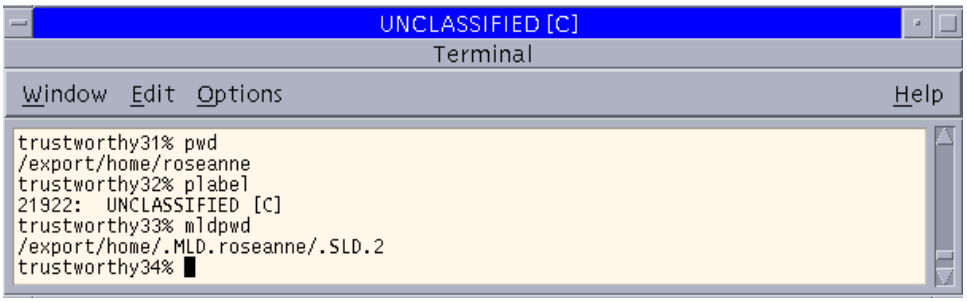



*Figure 2-19* Example SLD Name for the Third SLD Created in a Home Directory

Figure 2-20 shows the output of the pwd, plabel, and mldpwd commands in a terminal at the TOP SECRET sensitivity label. The TOP SECRET SLD was created fourth, so it has an SLD number of 3: .SLD.3.

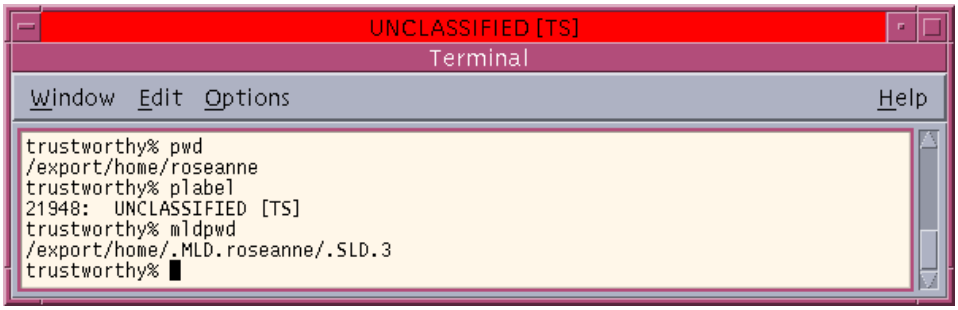



*Figure 2-20* Example SLD Name for the Fourth SLD Created in a Home Directory

### ▼ To Find Out if a Directory is an MLD

♦ **Use `getfattrflag(1TSOL)` and specify the directory name (without the MLD prefix).**

```
trusted% getfattrflag -m olddir
oldir: is a multilevel directory
```

### ▼ To Create an MLD from the File Manager

1. **From the File Menu, select New.**
2. **Enter the name of the new directory.**
3. **Click the button "Create New Directory as an MLD."**

### ▼ To Create an MLD from the Command Line

♦ **Use `mkdir` and specify the directory name with the MLD prefix.**

```
$ mkdir .MLD.newdir
```

OR

♦ **Use** `mkdir -M` **and specify the directory name (without the MLD prefix).**

```
$ mkdir -M newdir
```

OR (to change an existing directory to an MLD)

♦ **Use** `setfattrflag` **and specify the name of an existing directory.**

```
$ setfattrflag -m oldir
```

## ▼ To Identify an MLD

Several methods exist to identify a multilevel directory.

♦ **Use the** `file` **command on the command line.**

```
trustworthy16% file /export/home/roseanne
/export/home/roseanne:multi-level directory
```

♦ **Use** `mldrealpath` **from the command line.**

```
trustworthy15% mldrealpath /export/home/roseanne
/export/home/.MLD.roseanne
```

♦ **Use** `getfattrflag -m` **and specify the name of an existing directory.**

```
$ getfattrflag -m /tmp
/tmp: is a multilevel directory
```

### ▼ To Identify an SLD

♦ **From the command line, use** `getsldname(1TSOL)` **followed by the name of the MLD.**

```
trustworthy16% getsldname /export/home/roseanne
.SLD.2
```

♦ **Within the MLD, use** `mldpwd` **from the command line.**

### ▼ To Address the Entire MLD

♦ **Use ls to recursively list the MLD by specifying the MLD prefix:**

```
$  ls -R /export/home/.MLD.admin/.SLD*
```

What displays depends on the label of the current process. Referring to the entire MLD is especially useful when you want to use `tar` to copy the directory to a backup tape.

### ▼ To Change the MLD Prefix of a File System

1. **Assume the secadmin role, and do the rest of the steps in an ADMIN_LOW profile shell.**
2. **To change the MLD prefix on an already-created filesystem, find out the current MLD prefix, and then use** `setfsattr` **with the** `-m` **option followed by the new MLD prefix.**

```
$ getmldadorn /spare
.MLD.
$ setfsattr -m hidden. /spare
$ getmldadorn /export
hidden.
```

See the `setfsattr(1MTSOL)` man page.

3. **To use an alternate MLD prefix when creating a new filesystem, use** `newsecfs` **with the** `-m` **option.**

```
$ newsecfs -m hidden. /spare
hidden.
```

If you do not supply the -m option with an alternate MLD prefix, the default prefix applies to the newly created file system. See also the `newsecfs(1MTSOL)` man page.

## ▼ To Remove an MLD

**Note –** Before being able to remove an MLD, you need to remove every SLD within it and any contents contained within each SLD.

1. **Bring up a workspace at the highest sensitivity label in your clearance.**
2. **Do one of the following to remove all the SLDs and their contents that are in your account's sensitivity label range:**
   a. **In the File Manager, enter the adorned name of the MLD in the text entry field, and choose Go To from the File Menu.**
   b. **Choose Show Hidden Objects from the View menu.**
   Drag all the SLDs from the MLD into the trash or select them and choose Put in Trash from the Selected menu.

   Figure 2-21 shows the File Manager in the `/tmp` MLD, which has been accessed directly using the adorned name, `/.MLD.tmp`, and it shows that all the SLDs (`.SLD.1`, `.SLD.2`, and so on) are shown by choosing Show Hidden Objects from the View menu.

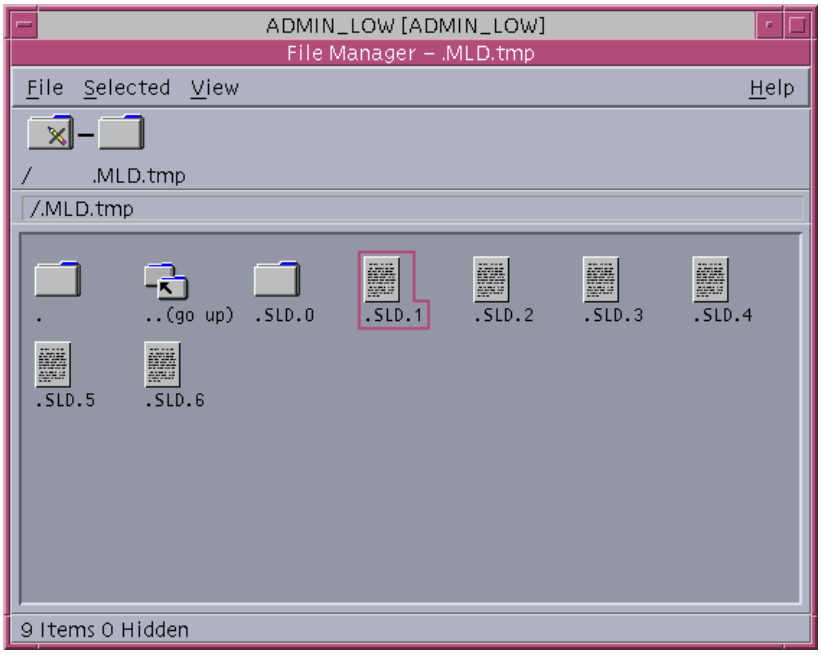



*Figure 2-21* Preparing the File Manger Before Deleting an MLD

**3. On the command line, use `rm` with the `R`, `M` and `f` options to remove the MLD, all the SLDs it contains and all of their contents.**

```
$ rm -RMf MLD_dirname
```

# 2

# *Part 2 — Administering Users, Roles, Profiles and Mail*

This part of the *Trusted Solaris Administrator's Procedures* manual contains six chapters that provide the needed background and the procedures for managing user and role accounts, mail, and printing.

Chapter 3, "Managing User Accounts" includes these topics:

| |
|---|
| *Decisions To Make Before Setting Up User Accounts* |
| *How Responsibilities for Managing Users Are Divided* |
| *Managing Startup Files in a Trusted Solaris System* |

Chapter 4, "Managing Roles" includes these topics:

| |
|---|
| *Differences Between Role Accounts and User Accounts* |
| *Differences Between Administrative and Non-Administrative Role Accounts* |
| *Dividing the Tasks of Managing User and Role Accounts* |
| *Authorizations for Access to Account Management Tasks* |
| *Alternatives to Two-Role Administration* |
| *Creating a New Role* |

Chapter 6, “Managing Mail” includes these topics:

| |
|---|
| *Changing Mail Aliases* |
| *Allowing Users to List the Entire Mail Queue* |
| *Tracing Sendmail’s Activities* |
| *Troubleshooting Mail Delivery Difficulties* |
| *Configuring Trusted Solaris Mail Delivery Options for Mail Below Users’ Minimum Labels* |
| *Substituting an Alternate Mail Application* |

Chapter 5, “Using the User Manager to Configure User and Role Accounts” includes these topics:

| |
|---|
| *Understanding the Information Entered in the User Manager Dialog Boxes* |
| *Setting Up or Modifying a User or Role Account* |

Chapter 7, “User Manager Data Collection Worksheet” provides a worksheet to collect data before you set up user and role accounts.

Chapter 8, “Managing Execution Profiles for Users and Roles” includes these topics:

| |
|---|
| *Background on Execution Profiles* |
| *Use of the Profile Manager to Create or Modify Execution Profiles* |

# *Managing User Accounts* 3

As described in *Trusted Solaris Installation and Configuration*, two user accounts are set up to assume the two administrative roles that work together as the "install team" during initial configuration of the Trusted Solaris system. This chapter describes how to finish essential tasks that are part of planning and setting up for all the remaining users on the system, which are not covered in the *Trusted Solaris Installation and Configuration* manual. This chapter also gives additional background needed for managing user accounts.


| | |
|---|---|
| *Things to Do Before Setting Up Accounts* | *page 58* |
| *Decisions To Make Before Setting Up User Accounts* | *page 58* |
| *How Responsibilities for Managing Users Are Divided* | *page 60* |
| *Managing Startup Files in a Trusted Solaris System* | *page 65* |
| *Administering the Automatic Running of Jobs Using cron, at, and batch* | *page 73* |


This chapter includes the following procedures:


| | |
|---|---|
| *To Make .login and .profile Looked at During Login for Bourne, C and Korn Shell Users* | *page 83* |
| *To Separate the Shell Initialization Files for Each Shell* | *page 83* |
| *To Propagate Startup Files to Home Directory SLDs* | *page 84* |

# 3

## Things to Do Before Setting Up Accounts

Set up user accounts after all these preconditions are met:

- All of the following are configured:
  - The NIS+ master
  - Any install and/or boot servers for the distributed system
  - Each user's primary workstation
  - Each user's home directory server
  - Each user's audit server(s)
  - Any other workstations and servers that provide any other services to the user's primary workstations, including the mail server
- Plan for all home directories for all users to be mounted from the NIS+ server.

**Note –** It is desirable for an account's home directory to be already mounted on the NIS+ master before you set up the account. If so, the User Manager will automatically set up the account's home directory as a multilevel directory (MLD) for you. If you do the set up of initialization files described under "To Separate the Shell Initialization Files for Each Shell" on page 83, the User Manager also populates the startup files from the skeleton path to the user's home directory SLD at the account's minimum sensitivity label.

## Decisions To Make Before Setting Up User Accounts

Make the following decisions before starting, because they affect how you configure accounts.

Some decisions are the almost the same as those you would make if installing a network of standard Solaris or other UNIX machines. However, because you are configuring the Trusted Solaris environment, you\ should consider these decisions in the light of any implications they might have for your site's security and of any special requirements you might have.

♦ **Decide on a convention for user login names**
The Trusted Solaris system enforces a minimum of six characters and a maximum of eight. Some organizations use first names followed by last initial; others use last names followed by first initial. Others have other conventions.

- **Review default groups and decide whether to add any groups**

  If you need to define new groups, use the Solstice Group Manager. No new fields have been added for Trusted Solaris group administration. Refer to the *Solstice AdminSuite 2.1 User's Guide*, if needed.

Make the following decisions and gather the specified details that are specific to Trusted Solaris user administration, based on your site's security policy.

- **Choose the overall method of password generation, automatic or manual**
- **Decide how your site wants to handle mail that is sent at an SL below the recipient's minimum SL.**
  Make sure that the Trusted Solaris-specific privacy options in the `sendmail` configuration file `sendmail.cf` have values consistent with your site's security policy. See "How Sendmail Handles Mail Below the Recipient's Minimum SL" on page 177 and "To Configure Mail Delivery Options for Mail Below Users' Minimum Labels" on page 178 in Chapter 6, "Managing Mail."
- **Have the following information available**:
  - A list of the available execution profiles

    The execution profiles shipped with the system are designed so that you may never have to change or extend them. See Appendix B, "Profile Definition Tables," for lists of which execution profiles are assigned to each role and their purposes, along with lists of all commands, actions, and authorizations assigned to each profile.

    If your site has special requirements, the security administrator can create more profiles. If other execution profiles exist at your site, see your own internal documentation for a description.
  - The available administrative and non-administrative roles.

    The roles shipped with the system are designed so that you may never have to change or extend them. If your site has special requirements, more roles may be added or the roles may be reconfigured. If your site has added or modified roles, see your own internal documentation for a description of the added or modified roles available for users at your site.

- The name of each of your site's employees who need accounts, along with the responsibilities, roles, clearances and minimum labels assigned to each.

Getting this information together may simply involve collecting data on the functions each employee performs in your organization and deciding the clearances, minimum labels, and profiles they should have. You also need to decide which employees are going to be able to assume administrative roles. At government organizations, you may need to review the government clearances that have been given to each employee, and use these to determine which roles they may assume and the labels at which they may work.

- **Decide whether to allow sourcing of shell initialization files and whether you want to control the initial contents of the files**
  See "Managing Startup Files in a Trusted Solaris System" on page 65.
- **Decide which files, if any, should be copied from the first $HOME directory SLD created for a user into subsequent SLDs, or whether they should be linked, and then modify the** `.copy-files`**, and** `.link-files` **in** `/etc/skel`**.**
  "To Propagate Startup Files to Home Directory SLDs" on page 84.

## *How Responsibilities for Managing Users Are Divided*

This section gives the background needed for managing user accounts under the following headings:


| | |
|---|---|
| *Managing Users: Divided Between Two Administrative Roles* | *page 61* |
| *System Administrator Responsibilities* | *page 61* |
| *Security Administrator Responsibilities* | *page 61* |
| *Alternatives to Two-Role Administration* | *page 62* |
| *Authorizations for Access to Account Management Tasks* | *page 62* |

## *Managing Users: Divided Between Two Administrative Roles*

In the default Trusted Solaris system, managing users is set up to be a two-person responsibility so that no single administrator has total control of the system. However, because more than one person may be enabled to assume any administrative role, administering users may be more accurately thought of a *two-role* responsibility than a *two-perso*n responsibility.

The tasks of managing user accounts are divided between these two roles:

- System Administrator
- Security Administrator

Authorizations that are configured into the default execution profiles for each of these two administrative roles control which fields in the account manager may be specified by each role. The authorizations that apply to the Trusted Solaris version of the Solstice User Manager are described under "Authorizations for Access to Account Management Tasks" on page 62.

## *System Administrator Responsibilities*

The system administrator role is authorized to specify all the aspects of the user account that are not security relevant, including

- User ID
- Group ID
- Home directory location, and
- The user's default shell

The tasks assigned to the system administrator role are the same as those performed by the system administrator (superuser or root user) in standard UNIX systems.

## *Security Administrator Responsibilities*

The security administrator specifies all security-related aspects of the user's account, including:

- The method of password generation
- The duration of password validity (minimum and maximum days before change is required, maximum days of inactivity, expiration date, warning)
- The user's minimum label

- The user's clearance
- The user's external or internal label view
- Whether the user sees CMW labels, SLs only or ILs only
- Whether the user is audited to a greater or lesser degree than all other users in the distributed system
- Which execution profiles the user has
- Which roles the user may assume
- What action to take, if any, when the workstation is idle for a specifiable amount of time

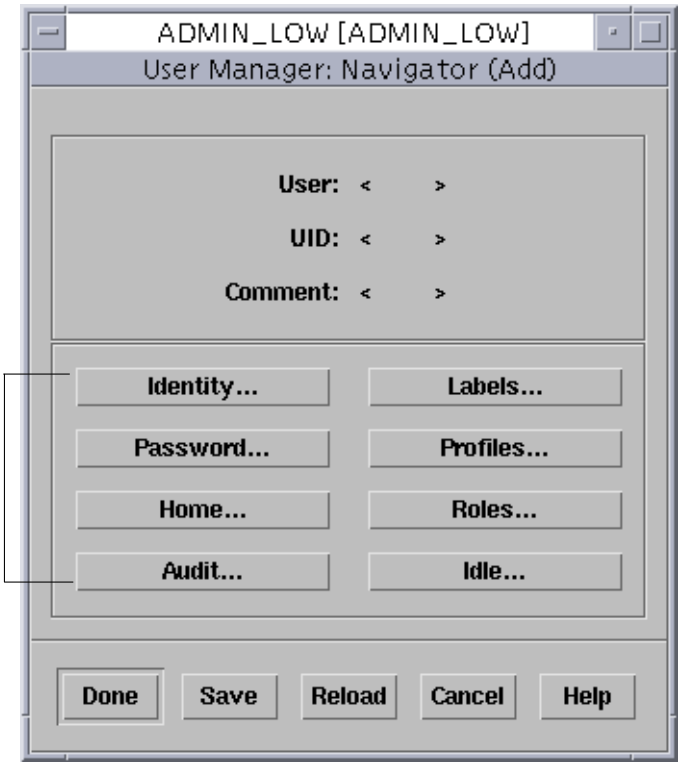
## *Alternatives to Two-Role Administration*

Your site may choose not to divide user administration between two roles and to reconfigure the administrative roles accordingly. See Chapter 4, "Managing Roles," under "Dividing the Tasks of Managing User and Role Accounts" on page 91.

**Note –** If administrative roles have been reconfigured at your site, the site's security administrator is responsible for informing administrators which responsibilities and capabilities have been assigned to each administrative role.

## *Authorizations for Access to Account Management Tasks*

Figure 3-1 on page 63 shows the User Manager Navigator (Add) menu for adding a user or role account. Because no information has yet been specified, the User Name, UID, and Comment fields are empty. The labels on the eight buttons below the Comment field indicate the types of information that is specified for each account. Each button brings up a dialog box in which the named type of account information is specified. These buttons appear on the User Manager: Navigator whether the navigator is being used to add or to modify a user account.




Buttons for access to dialog boxes where an administrative role specifies the named type of account information

*Figure 3-1* User Manager: Navigator

Each button requires an authorization. The authorizations are divided between the default system administrator and security administrator roles to maintain the separation of duties that is a requirement for two-person control of account administration.

---

**Note –** If any buttons are grayed out when an administrative role brings up the User Manager, this means that the authorization needed for that button is not in the role's execution profile.

---

The authorization required for each button, and the information specified on the corresponding dialog box is shown in Table 3-1.

*Table 3-1* Authorizations for User Manager Buttons and Types of Information Specified

| Button | Authorization Name | Default Role | Information |
|---|---|---|---|
| Identity | set user identity | sysadmin | • set account name, UID, primary and, supplementary GIDs, comment, shell, type of account: normal, admin role, non-admin role |
| Password | set user password | secadmin | • create a password for the account or set up account without a password<br>• set duration of password validity (min and max days before change required, max days of inactivity, expiration date, date for warning)<br>• set user's password generation method<br>• specify account state: open, closed<br>• choose whether or not to update NIS+ cred table |
| Home | set attributes related to home directories | sysadmin | • specify whether or not to create home directory at *pathname* on *server*<br>• specify path for shell initialization files<br>• set home directory permissions<br>• specify mail server<br>• choose whether or not to automount home directory |
| Labels | set user labels | secadmin | • set clearance and minimum label<br>• set external or internal label view (hide or show the names of administrative labels)<br>• specify viewing or hiding of SLs<br>• specify viewing or hiding of ILs |
| Profiles | set user profiles | secadmin | • for user accounts only: select user profiles |
| Roles | set list of assumable roles | secadmin | • for user accounts only: select one or more roles |
| Idle | set idle time | sysadmin | • specify action to take (if any) if screen is idle for the amount of time selected from the given list |

# Managing Startup Files in a Trusted Solaris System

In the Trusted Solaris system, because home directories are usually multilevel directories and because the profile shell may be assigned to an account to control access to commands, administrators setting up startup files must consider certain details that either are not as important or do not apply in standard UNIX systems. This section provides the information needed to understand how startup files are administered in the Trusted Solaris environment and provides procedures for doing the setup.

In UNIX systems, one set of startup files is read every time a user logs in while the window system is coming up, and another set of startup files is read whenever a user brings up a shell [`csh(1)`, `ksh(1)`, `pfsh(1MTSOL)`, or `sh(1)`] in a terminal emulator, such as the `cmdtool`, `shelltool`, or `dtterm`. The following discussion, "Controlling Which Startup Files Are Read While the Window System is Coming Up," only applies to the set of startup files that are sourced by the window system while it is being launched. See the discussion starting under "Skeleton Directories" on page 69 for how to administer the startup files that are read whenever a user brings up a `csh(1)`, `ksh(1)`, `sh(1)`, or `pfsh(1TSOL)`.

## Controlling Which Startup Files Are Read While the Window System is Coming Up

See Table 3-2 for a list of shell initialization files that are read when the user logs in, while the window system is coming up.

*Table 3-2* Startup Files Read by the Window System

| Shell | Startup File Read |
|---|---|
| C shell | `/etc/.login` and `$HOME/.login` |
| Bourne shell or Korn shell | `/etc/.profile` and `$HOME.profile` |
| Profile shell (only in Trusted Solaris systems) | `/etc/.profile` and `$HOME.profile` |

In the base CDE window system, every account gets an editable `$HOME/.dtprofile` file whose primary job is to control whether the `.login` or `.profile` files are read by the desktop when the account logs in and starts a session.

In the Trusted Solaris system, when any account (except one with the `pfsh(1MTSOL)` as its default shell) logs into the desktop, the window system copies either an `/etc/dt/config/sys.dtprofile` file that was created by the site's security administrator, if the file exists, or the default `/usr/dt/config/sys.dtprofile` into that account's `$HOME/.dtprofile`. See Figure 3-2.

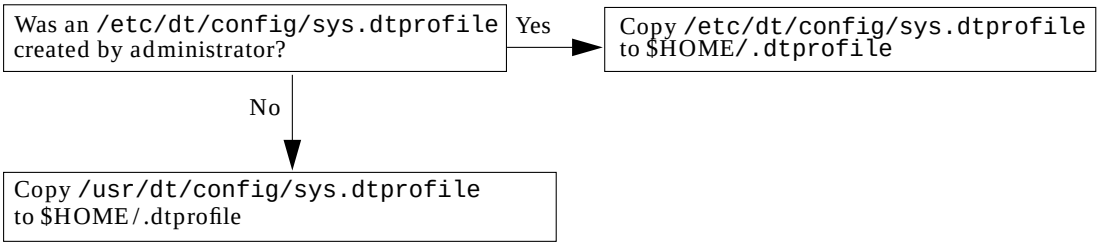



*Figure 3-2* How `$HOME/.dtprofile` is installed

As shipped, in `/usr/dt/config/sys.dtprofile`, the variable that enables the sourcing of either file is commented out (see Figure 3-3), and as a result the default action is to not source either file.

```
# DTSOURCEPROFILE=true
```

*Figure 3-3* Default Setting in the`/usr/dt/config/sys.dtprofile`

A `/etc/dt/config /sys.dtprofile` may be created and edited by the security administrator or the `$HOME/.dtprofile` may be edited by users with any other default shell than the `pfsh(1MTSOL)` to allow the sourcing of `.login` or `.profile` by the window system. Any of the `*.dtprofile` files may potentially be edited to do the same types of things done in other startup files, such as setting environment variables and search paths for commands and actions, and invoking commands or functions.

Comments in the *.dtprofile files encourage the site's administrator or individual users to modify the startup files before changing the default, while making sure that the startup files do not do anything that either requires a terminal emulator or requires user interaction while the window system is coming up. See the comments in the /etc/dt/config/sys.dtprofile file and "To Make .login and .profile Looked at During Login for Bourne, C and Korn Shell Users" on page 83, if changing the default is consistent with your site's security policy.

---

**Note –** If any modifications to a .login or .profile prevent the user from working, the user may use the Failsafe Session option on the login dialog. Failsafe Session allows a log in without the sourcing any startup files—to enable the user to fix the problem file.

---

## *How the Reading of Start Up Files Is Controlled for the Profile Shell User*

The algorithm for the profile shell is different from that applied with the other shells, for security reasons. The profile shell restricts the set of commands available to an account, and it accomplishes this by executing only the commands specified in one of that account's execution profiles. To support the special job of the profile shell, *when a user's login shell is specified as the Profile shell*, the .profile file *is* looked at during login, but that account's $HOME directory does not get an editable version of the .dtprofile file. The default in /usr/dt/config/sys.dtprofile or a version modified by the security administrator in /etc/dt/config/sys.dtprofile is used instead of $HOME/.dtprofile file.

---

**Note –** If the user whose default shell is pfsh creates a $HOME/.dtprofile, the setting in the personal .dtprofile can have no effect because it will never be looked at.

---

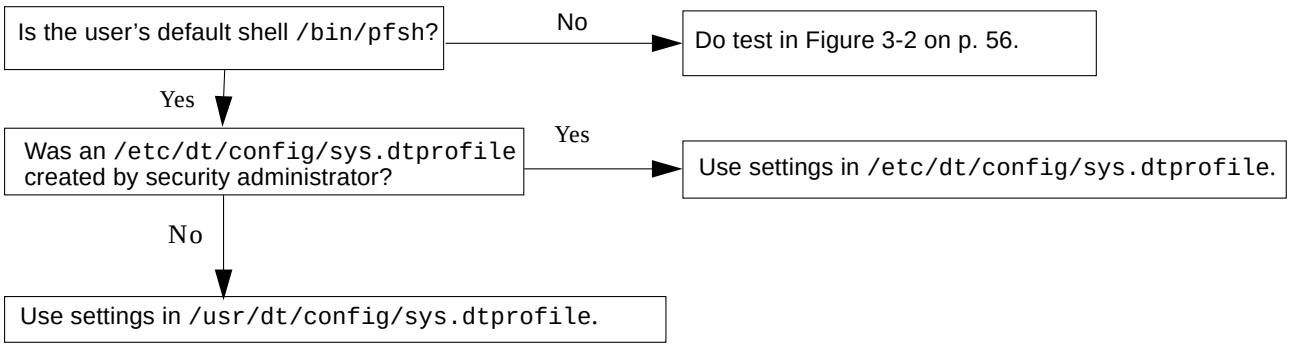



*Figure 3-4* How `$HOME/.dtprofile` is Bypassed for Users with a Default Shell of `pfsh(1MTSOL)`

## *Startup Files Launched When a Shell Comes Up,*

Shell initialization files usually set search paths and other environment variables and execute some useful commands and functions. Another startup file that is useful to have in a home directory is the `.mailrc`, which is often used to specify the user's Mail folder, inbox and mail aliases, among other things, as shown in Figure 3-5.

```
set folder=/home/roseanne/Mail

set MBOX=$HOME/Mail/inbox

alias pubs sharonv@think monicap@owl jstearns@auburn
```

*Figure 3-5* `.mailrc` Example

To give another example, the `.newsrc` file is consulted to determine which news groups to bring up whenever a user brings up a news viewer.

A short list of startup files for commonly-used applications is shown in Figure 3-6.

| Startup Files for Commonly Used Applications |
|---|
| `.cm.rc` |
| `.netscape` |
| `.mailrc` |
| `.newsrc` |

*Figure 3-6* Startup Files for Commonly Used Applications

## *Skeleton Directories*

If there is no administrative intervention and if the system administrator role specifies `/etc/skel` as the skeleton path directory when setting up an account, default initialization files for all the shells (`local.cshrc`, `local.profile`, and `local.login`) are copied into the account's $HOME directory from `/etc/skel`. The system administrator role may:

- Add to `/etc/skel` other files to be copied,
- Modify or rename the default files in `/etc/skel`.
- Specify an alternate skeleton directory that contains site-specific skeleton files

The system administrator can specify an alternate directory in the HOME dialog box when setting up a user's account using the Trusted Solaris version of the Solstice User Manager.

See "To Separate the Shell Initialization Files for Each Shell" on page 83 in this chapter for how to do the setup for all shells including the `pfsh`. System administrators may set up master copies of the shell startup files in one skeleton subdirectory for each shell and then specify in the User Manager the appropriate skeleton directory for each user's default shell. In this way, only the correct startup files for the user's $SHELL would be copied from the skeleton file directory into the first SLD created in the user's $HOME directory the first time the user logs in.

The copying of files from `/etc/skel` (or whatever skeleton path is specified) is done as it would be in other UNIX systems, but the fact that almost all home directories are MLDs complicate the process. In the Trusted Solaris system, the files are copied from `/etc/skel` *only* into the first SLD created on the account's behalf at the account's minimum (initial) label.

After initial login, while in the home directory SLD at the account's minimum label, each account is responsible for renaming the file or files copied from the skeleton directory that are appropriate to his or her default shell and modifying the files to suit their needs. For example, if a user's default shell is a C shell and minimum (initial) sensitivity label is PUBLIC, after the first login, in the PUBLIC SLD in the home directory, the user would rename the `local.cshrc` automatically copied from `/etc/skel` to `.cshrc`, and modify the `.cshrc` as desired.

Because the initialization files are only loaded into the first home directory SLD created at the accounts's minimum label, more work needs to be done to propagate the files to any SLDs that may be created at other labels. For any initialization files to be copied or linked into other home directory SLDs created at other labels, either the user or administrator needs to create either a `.copy_files` or `.link_files` or both, as described beginning in "Using .copy_files and .link_files" on page 70.

## *Using* `.copy_files` *and* `.link_files`

Two new Trusted Solaris files `.copy_files`, and `.link_files` can now be put into the first home directory SLD created at any account's initial sensitivity label to help automate the copying or linking of any startup files into any subsequent SLD created within a user's home directory. The user or the administrator can list in `.copy-files` whatever files should be copied and list in `.link-files` whatever files should be linked into any subsequently created SLD(s) from the first SLD. Zero or more of either of these files can be created in the first home SLD. Whether the files are copied or linked is at the discretion of whoever is doing the setup.

### *If* `.copy_files` *is Used to Copy Files Between SLDs:*

Each copy may be modified to be different in each SLD, if desired. After any file is copied from the first SLD into a subsequent one, the file may be edited by the user, so that different versions may appear in different SLDs. Copying

would be desirable, for example, if users need to use different mail aliases when they are working at different sensitivity labels. To put a copy of the .`mailrc` in each SLD after the first one, you would include the name of the `.mailrc` in the `.copy_files` file. See "To Propagate Startup Files to Home Directory SLDs" on page 84.

Following these steps, the system administrator role creates a skeleton directory (or uses the default `/etc/skel`) and puts into it generic copies of startup files. The administrator also creates in the skeleton directory a master `.copy_files` containing a list of any startup files to be copied and/or a master `.link_files` containing a list of any startup files to be linked to all home SLDs. When setting up user accounts, the system administrator specifies the pathname of the modified skeleton directory. The result of this setup is that all the files from the skeleton directory (including the `.link_files` and/or `.copy_files`) are copied into the user's initial home SLD, and, based on the `.link_files` and/or `.copy_files`, the specified files are either linked or copied to every subsequently-created SLD.

### *If* `.link_files` *is Used to Link Files Between SLDs:*

A change made to one initialization file in one SLD is made to all the files at all labels in all SLDs to which the file is linked.

So, for example, if a change is made to the linked `.cshrc` file in a Confidential workspace, the change would apply to all the `.cshrc` files in all other SLDs at all other labels. See "To Propagate Startup Files to Home Directory SLDs" on page 84.

## *Worksheet for Copy and Link Files*

Here are some examples of common files with a worksheet for planning which files to copy or link.

*Figure 3-7* Planning Worksheet for Copying and Linking Startup Files Between SLDs

| **Common Startup Files** | **List to be Copied (for** `.copy_files`**)** | **List to be Linked (for** `.link_files`**)** |
|---|---|---|
| `.bugtraqrc` | | |
| `.cm.rc` | | |
| `.cshrc` | | |
| `.dtprofile` | | |
| `.login` | | |
| `.netscape` | | |
| `.mailrc` | | |
| `.newsrc` | | |
| `.profile` | | |

# *Administering the Automatic Running of Jobs Using* cron, at, *and* batch

cron(1MTSOL is the clock daemon that executes commands at specific times.

This section gives a brief overview of what cron does and describes how administering cron and its associated commands is different in the Trusted Solaris system. See the Solaris 2.x *System Administration Guide, Vol. II* for basic cron information. For Trusted Solaris modifications see also the modified man(1) pages for at(1TSOL), atq(1TSOL), atrm(1TSOL), batch(1), cron(1MTSOL), and crontab(1TSOL).

## *Review of Concepts*

cron maintains an internal time-ordered list of events for all scheduled jobs. Each event represents a job with the information necessary to execute it. cron runs two types of jobs:

- Single execution jobs (at_jobs) scheduled by at to be executed once at a specified time in the future
- Single execution jobs (at_jobs) scheduled by batch, which is a front-end script to at(1) that submits jobs to be executed right away, as soon as the system load level permits
- Periodic execution jobs (cron_jobs) scheduled by crontab, to be executed repetitively at specified intervals

cron_jobs and at_jobs are scheduled by cron from reading crontab and atjob files in their respective spool directories only at the following times:

- During cron's own process initialization or
- After change is made to a crontab or an atjob file.

### crontab *Files*

The crontab file is generated by a user or role account using the crontab(1TSOL) command (which, in the Trusted Solaris system, must be in one of the account's execution profiles). A crontab file consists of commands, one per line, that execute automatically at the time specified at the beginning of each command line. Each command line is referred to as a *cron_job*. A crontab file may contain multiple cron_jobs. The spool area for the crontab files is /var/spool/cron/crontabs.

### `atjob` *Files*

The `atjob` file is generated by a user or role account using the `at(1TSOL)` or `batch(1TSOL)` command (either of which, to be used in the Trusted Solaris system, must be in one of the account's execution profiles). The user's current process environment when the command is executed is saved as part of the file. Each file is referred to as an *at_job*. The spool area for the `atjob` files is `/var/spool/cron/atjobs`.

### *Supporting Jobs at Multiple Labels in the Spool Directories*

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

## *Determining Whether the Profile Shell is Used By a Job*

**Caution –** If the profile shell, `pfsh`(1TSOL), is specified to execute a job, the security administrator must ensure that all of the job's commands are also in an execution profile assigned to the invoking user.

cron_jobs are executed using `pfsh` if either of the following is true:

- The login shell in the account's `passwd`(4) entry is the `pfsh` or
- The $SHELL environment variable is the `pfsh`

Otherwise, `cron` uses the default Bourne shell, `sh(1)`, for cron_jobs.

Because a user can use `at` with the -c (for `csh`), -k (for `ksh`), -s (for `sh`), or `-p` (for `pfsh`) options to specify the shell with which the job should be run, for at_jobs there is a third case in which the profile shell is used. at_jobs are executed in the profile shell if either:

- The login shell in the account's `passwd` entry is the `pfsh` or
- The $SHELL environment variable is the `pfsh` or
- The `at` command is specified with the `-p` option

Otherwise, `at` uses:

- Any shell specified with either the `-c`, `-k`, or `-s` options or
- The default shell, sh

## *Running Privileged Commands in* `at` *or cron Jobs*

If a command in an `at` or `cron` job needs to run with privileges, either forced or inheritable privileges may be made available.

Allowing a command to run with forced privileges no matter who executes the command is not usually consistent with a site's security policy, so the security administrator usually needs to do the following to make the privileges available by inheritance:

- Specify the command and any privileges it needs in one of the invoking user's execution profiles using the Profile Manager and
- Specify that the job is executed with the profile shell, as described in "Determining Whether the Profile Shell is Used By a Job" on page 74

See "How Privileges Are Assigned to Commands and Actions" on page 453 and "To Give Forced Privileges to a Command" on page 476, or "Giving Inheritable Privileges to a Command or Action" on page 454 for more information. See also, "To Write a Profile Shell Script That Runs Privileged Commands" on page 479, which uses a cronjob in the example.

## *Using a UNIX Domain Socket for Communications*

The communication mechanism between `crontab(1TSOL)`, `at(1TSOL)`, `atrm(1)` and `cron(1MTSOL)` in the Trusted Solaris system is a UNIX domain socket. Using the UNIX domain socket, the trusted networking interfaces provided by the Trusted Security Information Exchange (TSIX) library allow `cron`'s clients to communicate their message and security attributes to `cron` in a trusted way. See also the man(1) page for `libt6`(3NTSOL).

`cron`(1MTSOL) is modified to create and bind the UNIX domain socket 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.

The clock daemon turns on the net_mac_read privilege in its effective set to make a multilevel port, so it can receive messages at different sensitivity labels.

When a user creates, modifies, or removes a `crontab` or `atjob` file, the `crontab(1TSOL)`, `at(1TSOL)`, or `atrm(1)` command sends a message (filled in with the appropriate data) along with the command's processes' sensitivity label to notify the clock daemon. Upon receipt of a message and the associated sensitivity label, the clock daemon interprets the message and performs the necessary operations. The submitting processes' sensitivity label is used to create or remove the `crontab` or `atjob` file at the corresponding SLD.

## *Ancillary Files*

An ancillary file is created in the `crontabs` MLD for each `crontab` file and in the `atjobs` MLD for each `atjob` file. Modification of crontab or 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.:

## Access to `at` *and* `cron`

Administrators can control access to `at` and `cron` by means of `*.allow` and `*.deny` files in the `/etc/cron.d` directory.

The two files that define the users permitted or forbidden to use `at` are:

- `at.allow`

- `at.deny`

The two files that define the users permitted or forbidden to use `cron` are:

- `cron.allow`
- `cron.deny`

If the `*.allow` file exists, then the `*.deny` file is not checked to determine the user's access. If no `*.allow` file exists, the `*.deny` file is checked, and if it exists but is empty then anyone is permitted to submit jobs. In Trusted Solaris, if neither an `allow` or `deny` file exists, no user is allowed to submit a job.

The Trusted Solaris system is delivered without `cron.allow` and `at.allow` files. The default `cron.deny` and `at.deny` files contain the following account names:

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

## *Allowing Access to Jobs Owned by Others*

The default Trusted Solaris security policy does not allow users to access jobs owned by other users, but the security administrator can configure users to bypass this restriction. To allow certain users to access jobs belonging to other users, the security administrator can use both of the following together:

- The `at.admin` and `cron.admin` files in `/etc/cron.d`
- Assigning `at` and `cron`-related authorizations

# 3

## `at.admin` *and* `cron.admin` *Files*

As shipped, the `at.admin` and `cron.admin` files in `/etc/cron.d` contain the names of special system accounts that are used by system processes. The default list, which is the same in both files, is as shown here:

```
bin
adm
lp
smtp
uucp
nuucp
listen
```

As shipped, the default `crontabs` ADMIN_LOW SLD contains `crontabs` for adm, sys, and uucp. There are no other default crontabs in the `crontabs` MLD or default at_jobs in the `atjobs` MLD. Any site can add additional crontabs and can also create at_jobs to run on the behalf of any of the special system accounts. Because no one logs into those accounts, which have no passwords assigned, creation of or any kind of modification to cron_jobs or at_jobs run on behalf of these accounts would otherwise not be possible without this feature.

## *Conditions for Access to Other's Jobs*

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

### *Conditions for* `at`*-related Commands*

When using `at`, `atq`, or `atrm`, for an account to create or access an at_job owned by another user:

1. The specified *username* or the *username* of the specified at_job's owner is one of the special system account names listed in the `at.admin` file and 3 must be true, or
2. The *username* of the specified at_job's owner is the name of a role account and 3 must be true.
3. account has the *modify at admin* authorization in an execution profile.

4. If neither of 1 or 2 is true, the invoking account must have the *modify at user* authorization in an execution profile

### *Conditions for the* `crontab` *Command*

When using `crontab`, for an account to create or access a `crontabs` file owned by another user:

1. The specified *username* is one of the special system account names listed in the `cron.admin` file and 3 must be true, or
2. The specified *username* is one of the role account names and 3 must be true.
3. The invoking account has the *modify cron admin* authorization.
4. If neither of 1 or 2 is true, the invoking account must have the *modify cron user* authorization in an execution profile.

# 3

## *Changes to* `crontab`*(1TSOL)*

The Trusted Solaris modifications to the crontab(1TSOL) command are described in Table 3-3.

*Table 3-3* **crontab(1TSOL) Options**

| Option | Comments |
|---|---|
| e | Create or modify a `crontab` file at the sensitivity label that matches the invoking processes' sensitivity label. A user can edit another user's crontab file only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 77 are met.<br>If the user's `passwd` entry does not specify `/bin/pfsh`, the `crontab -e` command invokes the editor defined by the VISUAL environment variable. If VISUAL is null, the editor defined in the EDITOR environment variable is used; if neither is defined, `cron` uses the default editor `ed`(1). When the user's `passwd` entry specifies `/bin/pfsh`, if the environment variable is set to `vi`, then `adminvi` is used. If the environment variable is set to be `dtpad`, then the `TSOLdtpad` editor is used. If neither variable is set, `cron` uses the default editor `adminvi`. |
| l | Display the `crontab` file at the invoking processes' sensitivity label for the current user. A user can display another user's `crontab` file only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 77 are met. |
| r | Remove a user's `crontab` file at the invoking processes' sensitivity label from the `crontabs` directory. A user can remove another user's `crontab` file only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 77 are met. |

The access control remains the same except that when neither `cron.allow` nor `cron.deny` exists, no user is allowed to submit a job.

## *Changes to the* `at` *Command*

Table 3-4 shows modified standard `at` options and the new `-p` option for `at`.

*Table 3-4* **Trusted Solaris 2.5** `at`**(1) options**

| Option | Comments |
|---|---|
| l | Display information about all at_jobs owned by the current user at the invoking processes' sensitivity label, if no at_job number operands are specified. If at_job number operands are specified, report information only for those jobs. If the at-job is not owned by the current user, display its job information only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 77 are met. |
| r | Remove the at_jobs with the specified at_job number operands that were previously scheduled. If the specified at-job is not owned by the current user, it is removed only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 77 are met. |
| p | The profile shell is used to executed the job. |

The access control remains the same except that when neither `at.allow` nor `at.deny` exists, no user is allowed to submit a job.

## *Changes to the* `atq` *Command*

Table 1-5 shows `atq` changes.

*Table 3-5* **Trusted Solaris 2.5** `atq`**(1) Changes**

| Argument | Comments |
|---|---|
| none | Display at-jobs owned by the invoking user at the invoking processes' sensitivity label. Other user's at-jobs are also displayed only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 77 are met. |
| *user* | Display all at-jobs owned by the specified *user*, if *user* is the invoking user. If *user* is not the invoking user, display jobs only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 77 are met. |

# 3

## *Changes to the* `atrm` *Command*

Table 1-5 shows `atrm` changes.

*Table 3-6* **Trusted Solaris 2.5 `atrm`(1) Changes**

| Argument | Comments |
|---|---|
| *user* | Remove all at-jobs owned by the specified *user* at the invoking processes' sensitivity label, if *user* is the invoking user. If *user* is not the invoking user, remove jobs only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 77 are met. |
| *at_job #* | Remove at_jobs with the specified at_job number operands at the invoking processes' sensitivity label. |
| *option* **a** | Remove all at-jobs owned by the invoking user at the invoking processes' sensitivity label. |

## *Miscellaneous*

`cron(1M)` is started at ADMIN_LOW sensitivity label by the boot profile, and then it is changed to run at ADMIN_HIGH sensitivity label after it creates the UNIX domain socket at ADMIN_LOW.

Trusted Solaris 2.5 is delivered with the following crontab files:

At the ADMIN_LOW sensitivity label, pairs of crontab and ancillary files for root, uucp, adm, and sys.

At the ADMIN_HIGH sensitivity label, pairs of crontab and ancillary files for root, and lp.

The `/var/cron/log` file is created by clock daemon at ADMIN_HIGH sensitivity label. The clock daemon logs its internal messages in this log file.

## *User Setup Procedures*

### ▼ To Make `.login` and `.profile` Looked at During Login for Bourne, C and Korn Shell Users

1. **To change the default for all users on the local host, assume the security administrator role**
2. **In the ADMIN_LOW workspace in a terminal, shell or command tool, make a copy of `sys.dtprofile` and open it for editing.**

   a. **Copy `sys.dtprofile` from `/usr/dt/config` to `/etc/dt/config`.**

```
$ cd /usr/dt/config
$ cp sys.dtprofile /etc/dt/config
```

   b. **Use the Admin Edit action to open the `sys.dtprofile` file for editing.**

3. **Remove the pound sign (#) before the DTSOURCEPROFILE variable assignment at the end of the file.**
   After editing, the line should look like this sample screen.

```
DTSOURCEPROFILE=true
```

4. **Save and close the file.**
5. **Reboot the host.**

### ▼ To Separate the Shell Initialization Files for Each Shell

- **In the system administrator role, follow the steps under "How to Set Up the User Initialization File" in *User Accounts, Printers and Main Administration* manual for Solaris 2.5.**
  The procedure tells you how to create three shell-specific directory names that you will enter in the Skeleton path field in the User Account Manager. 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.

- **As admin, create a `skelP` subdirectory (for a specialized version of the `.profile` to be installed into the home directories of users whose default shell is `pfsh`).**
- **As admin, enter the correct `skel`*X* subdirectory name into the Skeleton path field in the User Manager, based on the user's default shell.**

## ▼ To Propagate Startup Files to Home Directory SLDs

**Note –** Any user can put a `.copy_files` or `.link_files` into his or her initial home SLD or modify them if they are already present by administrative action

**Warning –** Do any setup for the startup files immediately, before users have a chance to work at multiple sensitivity labels. Otherwise, multiple home SLDs may be created without startup files being propagated.

1. **Assume the system administrator role.**
2. **In an ADMIN_LOW workspace, go to `/etc/skel` or whatever skeleton directory location you wish to use.**
   Figure 3-8 shows a skeleton directory created for startup files for the C shell, `/etc/skel/skelC` (according to the procedure "To Separate the Shell Initialization Files for Each Shell" on page 83).

```
$ cd /etc/skel/skelC
```

*Figure 3-8* Changing to a Skeleton Directory Created for C Shell Startup Files

3. **Put generic copies of startup files into the skeleton directory.**
   Figure 3-9 shows startup files for the C shell, for the calendar manager, for the mailer and for an Internet browser

```
$ ls
.cm.rc
.cshrc
.login
.mailrc
.netscape
```

*Figure 3-9* Startup Files in `/etc/skel/skelC`

4. **Create a `.copy_files` or modify it, if it already exists, to list any files you want to have copied to all home directory SLDs.**
   Remember that copied files may be different in each SLD.

5. **Create a `.link_files` or modify it, if it already exists, to list any files you want to have linked to all home directory SLDs.**
   Remember that all linked files are identical in each SLD.

6. **When setting up user accounts, enter the pathname of the modified skeleton directory in the Home dialog box on the User Manager.**

# 3

# *Managing Roles* 4

This chapter gives the necessary background and describes how to create and modify roles under the following headings:


| | |
|---|---|
| *Differences Between Role Accounts and User Accounts* | *page 88* |
| *Differences Between Administrative and Non-Administrative Role Accounts* | *page 88* |
| *Non-administrative Roles* | *page 88* |
| *Administrative Roles* | *page 89* |
| *Dividing the Tasks of Managing User and Role Accounts* | *page 91* |
| *Authorizations for Access to Account Management Tasks* | *page 91* |
| *Alternatives to Two-Role Administration* | *page 94* |
| *Required Privileges* | *page 95* |
| *Override Privileges* | *page 96* |
| *Creating a New Role* | *page 95* |
| *To Configure a New Role* | *page 98* |

# 4

## Differences Between Role Accounts and User Accounts

Role accounts have all the characteristics of user accounts, with the following exceptions:

- The system does not allow role account to log in.

  Administrative work must be done only by known users who have previously identified themselves to the system and who have been authenticated. This requirement is based on the need for all users and especially for administrators to be accountable for their actions. Administrative actions performed by a user whose identity is not known are not attributable through the auditing mechanism.

- Instead of logging into a role, already-identified and authenticated users assume a role from the Trusted Path menu.

- A role cannot assume another role.

  If you identify an account as a role account in the User Manager Identity dialog box, the Role button dims on the Navigator dialog box and if you then click on the Role button you get an error message.

- Each role has a role workspace that gets added to the front panel when the user assumes the role.

## Differences Between Administrative and Non-Administrative Role Accounts

Two types of roles are available:

- Non-administrative
- Administrative

### Non-administrative Roles

Only one non-administrative role exists in the default system:

- System Operator

The operator role exists to do backups and manage printers, and none of its functions requires the Trusted Path Attribute or any of the other special characteristics described under “Administrative Roles.”

### *When to Create a Non-administrative Role*

When deciding whether to create a non-administrative role compared to simply creating a new non-role execution profile and assigning it to users, you should think about the advantages of creating the non-administrative role described here.

Non-administrative roles are able to share ownership of and access to files between multiple users, and all users who can assume the same role are able to share the same environment in a single shared workspace:

- Any files created by the role are owned by the role, so any employee allowed to assume the role can have the same access to the files as the creator.
- Everyone who assumes the role gets the same home directory and environment set up in the role workspace.

For example, you can set up several employees who are able to assume the operator role (account name *oper*) on various shifts. Each user acting in the operator role has the same home directory ($HOME/`oper`), and can use the same tools. Any files created by the role are owned by `oper`.

## *Administrative Roles*

Two main administrative roles in the default Trusted Solaris system are:

- Security administrator
- System administrator

A third optional administrative role, `root`, exists for installing software or doing other actions where a real UID of 0 (root's UID) is required. The root role should ordinarily be assigned to the same account that has the secadmin role, because while root's capabilities are strictly limited, the root role's main function of adding software is security-sensitive

Administrative roles have the following unique characteristics:

- An administrative role is automatically assigned the sysadmin group 14 as one of its supplementary groups. Without this group ID the role cannot run administration tools.
- The administrative role workspaces have the Trusted Path Attribute that is also required for using the administration tools.

## When to Create a New Administrative Role

Sites may create a new administrative role if they have a need for an additional role to use commands or applications that check for the Trusted Path Attribute. A site might create a new administrative role, for example, if it wanted to combine the responsibilities of the three default roles into one. See “Dividing the Tasks of Managing User and Role Accounts” on page 91.

## Things That Need the Trusted Path Attribute

Table 4-1 shows the commands and applications that require the Trusted Path Attribute, and which therefore must be run in an administrative role’s workspace.

*Table 4-1* Commands and Applications Requiring the Trusted Path Attribute

| |
|---|
| `audit(1MTSOL)` |
| `auditd(1MTSOL)` |
| `login(1)` `-d` and `-f` options |
| `runpd(1MTSOL)` |
| `sendmail(1MTSOL)` options `-ba`, `-bd`, `-bi`, `-bd`, `-bs`, `-bt`, `-bv`, `-M`, `-X`, `-d`, and `-q` |
| Solstice AdminSuite tools: |
| Database Manager `dbmgr` |
| Group Manager `groupmgr` |
| Host Manager `hostmgr` |
| Printer Manager `printmgr` |
| Profile Manager `prfmgr` |
| Serial Manager `serialmgr` |
| User Manager `usermgr` |

For example, the Solstice AdminSuite tools, which are used for most administration tasks, check for the Trusted Path Attribute, so if the security administrator creates a new role that needs to access any of the Solstice AdminSuite tools listed in Table 4-1, it would need to be created as an administrative role.

## *Dividing the Tasks of Managing User and Role Accounts*

As described in Chapter 3, “Managing User Accounts,” in the default Trusted Solaris system, the security administrator and system administrator roles each configure a prescribed list of characteristics for user accounts. As illustrated in Figure 4-1, the security administrator role is also responsible for configuring the security relevant characteristics of normal user accounts and the accounts for all other roles, while, because no role can configure itself, the system administrator is responsible for configuring the same characteristics of the security administrator role as he or she configures for user accounts.

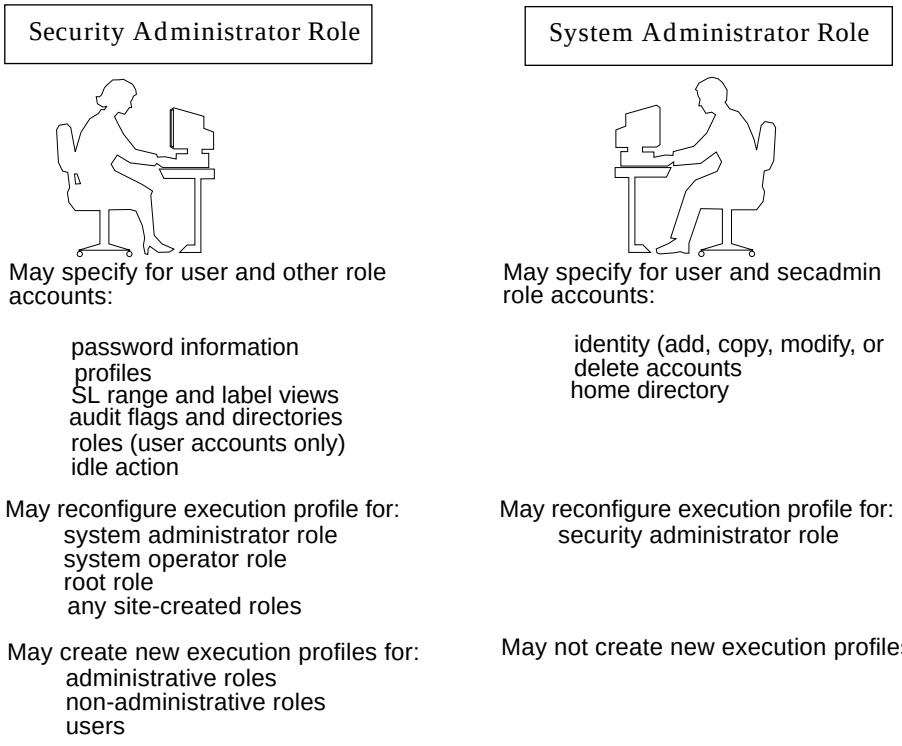



*Figure 4-1* Division of Account and Profile Configuration Responsibilities Between Security Administrator and System Administrator

# 4

## *Authorizations for Access to Account Management Tasks*

Figure 4-2 shows the User Manager Navigator (Add) menu for adding a user or role account. Because no information has yet been specified, the Account Name, UID, and Comment fields are empty. The labels on the eight buttons below the Comment field indicate the types of information that is specified for each account. Each button brings up a dialog box in which the named type of account information is specified. These buttons appear on the User Manager: Navigator whether the navigator is being used to add or to modify a user account.

Buttons for access to dialog boxes where an administrative role specifies the named type of account information

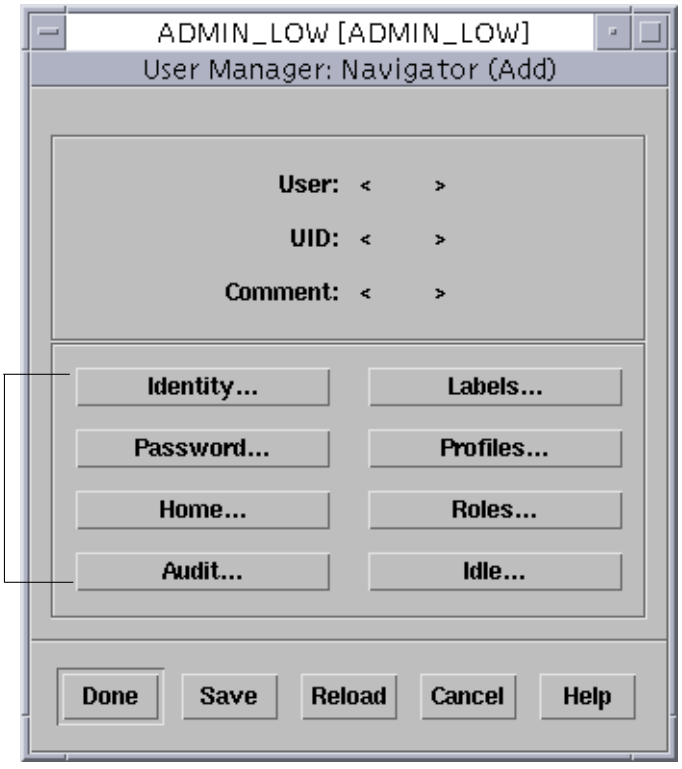



*Figure 4-2* User Manager: Navigator

Each button requires an authorization. The authorizations are divided between the default system administrator and security administrator roles to maintain the separation of duties that is a requirement for two-person control of account administration.

**Note –** If any buttons are grayed out when an administrative role brings up the User Manager, this means that the authorization needed for that button is not in the role's execution profile.

The authorization required for each button, and the information that can be specified on the corresponding dialog box is shown in Table 4-2. See also the `auth_desc(4TSOL)` man page.

*Table 4-2* Authorizations For Specifying Types of User Information

| User Manager Button | Authorization Name | Number | Scope of Authorization |
|---|---|---|---|
| Identity | set user identity | 17 | Allows an administrator to set security information related to the user's identity via the User Manager: the user name, primary group, secondary groups, comment, and login. Needed to add, copy, or delete a user. |
| Password | set user password | 18 | Allows an administrator to set password information pertaining to a user via the User Manager, including the password, type of password, life time, expiration date, warning days, and the permission to set up the credentials table. |
| Home | set attributes related home directories | 25 | Allows an administrator to determine such things as location, permissions, and initial contents of a user's home directory via the User Manager. |
| Labels | set user labels | 20 | Allows an administrator to set various label-related pieces information associated with a particular user via the User Manager: the user's minimum login label, clearance, label view, and label translation attributes. |
| Profiles | set user profiles | 22 | Allows an administrator to assign profiles to a user via the User Manager. |
| Roles | set list of assumable roles | 24 | Allows an administrator to select via the User Manager which roles a user may assume. |
| Idle | set idle time | 23 | Allows an administrator to set the idle time and determine which command to execute when a workstation has been idle for the specified idle time via the User Manager. |

**Note –** It is important to understand that the default authorizations shown in Table 4-2 on page 93 allow the role to modify the appropriate information for every other user and role account except that of the role itself. For example, the set user password authorization allows the security administrator to configure password options for all other accounts except that of the security administrator role.

## *Authorization for Specifying Information for One's Own Role*

The *permit self-modification* authorization allows a role who has that authorization to specify the authorized information for the role's own account.

By default, neither of the administrative role profiles have this authorization.

The permit self-modification authorization allows a role only to partially bypass the restriction on modifying one's own role. This authorization therefore actually allows the role to configure for itself only those areas where the role already has the required authorizations. For example, if the system administrator adds the permit self-modification authorization to the execution profile for a security administrator—who has the authorizations that give access to the security relevant dialog boxes in the User Manager, the security administrator role could then only specify the security-relevant aspects of its own account, and still would not be able to enter information in any other fields in the dialog boxes for which the role was not authorized, such as those on the Identity dialog box.

**Note –** The permit self-modification authorization does not allow a role to edit any profiles assigned to that role account.

## *Alternatives to Two-Role Administration*

Your site may choose not to use two role administration and may decide to create a single combined administrative role to do administrative tasks, if your site's security policy allows.

**Note –** Do not attempt to combine the capabilities of secadmin and sysadmin in the root role because the root role cannot be added to the NIS+ admin group, and the root role cannot run Solstice from any other host than the NIS+ master. As a result, a combined root role would not be able to run any of the Solstice administration tools unless directly logged into a the NIS+ master.

**Note –** If your site wishes to have a single combined administrative role, make sure that the combined role includes all authorizations that are in both role's profiles along with the authorization *permit self modification*, which is described in "Authorization for Specifying Information for One's Own Role" on page 94. You may also wish to add the commands, authorizations, and actions assigned to the system operator role.

## *Creating a New Role*

In Trusted Solaris 2.5 system, each role must have its own individual account, so the system administrator and security administrator must set up an account for a new role. The security administrator may also need to create a new profile for the role.

Before creating a new profile for a role, the security administrator should analyze whether any of the commands required by the role need privileges in order to do the tasks the role is assigned to do. See Chapter 16, "Adding Software," and `priv_desc(4TSOL)`. See the man pages for guidance about how to determine *required* privileges and *override* privileges that may be required by the command.

### *Required Privileges*

When a man page that has not been modified for the Trusted Solaris system states that super-user is required to execute a certain command or option, remember that one or more privileges are required instead. When a command or one of its options needs a privilege in order to succeed, that privilege is a *required* privilege, which is sometimes called an *operational* privilege. Required privileges are indicated on a Trusted Solaris man page with the words "must have" as shown in this sentence: "The `ifconfig(1MTSOL)` command must

have the sys_net_config privilege to modify network interfaces." If the required privilege is not given to the command in a user's execution profile by the security administrator, the command will not work.

## Override Privileges

*Override* privileges are needed for a Trusted Solaris command that is designed to work within security policy when the command fails because certain DAC or MAC checks are not passed. Like required privileges, override privileges may be assigned at the security administrator's discretion, after making sure that the override privileges do not give the user powers beyond his or her level of trust or allow actions beyond the scope of the task for which the profile is designed. On man pages, the names of privileges that may be used to override access restrictions are given in the ERRORS section. The override privileges that may be given to bypass DAC or MAC restrictions on files or directories are given below.

### DAC Override Privileges

The DAC override privileges are file_dac_read and file_dac_write. If a user does not have DAC access to a file, the security administrator may assign one or both of these privileges to a command, depending on whether read or write access or both are desired.

### MAC Override Privileges

The MAC override privileges are file_mac_read and file_mac_write. If a user doesn't have MAC access to a file, the security administrator may assign one or both of these privileges to the command, depending on whether read or write access or both are desired.

### Options for Avoiding the Need for Privilege

Besides being able to assign an override privilege, the security administrator has other options. For example, to avoid the use of privilege the security administrator may specify that the command executes with another user's ID (usually the root ID 0) or group ID, one that allows access to the file or directory based on its permissions or its ACL.

Also, when configuring an administrative role, the security administrator needs to decide whether a command needs to run at a certain label.

### *Verifying the Use of Security Attributes Within Security Policy*

After assessing which privileges and other security attributes a command needs in order to do its work, the security administrator needs to ascertain whether the command and the role may be trusted to use the privileges and other security attributes without violating the site's security policy.

### *Example: Using the Man Page When Configuring* `mount` *in a Profile*

The `mount(1MTSOL)` command is specified in the System Maintenance profile assigned to the system administrator role. It is a fairly-simple example of how specifying security attributes is done. In Solaris and other UNIX operating systems, `mount` without any options displays the name of the mounted filesystems to an ordinary user. However, when run as root, `mount` may be used by the administrator (super-user) to mount filesystems from the command line or to mount filesystems specified in the `/etc/vfstab` table. The `mount(1M)` command checks for UID=0.

In Trusted Solaris, `mount` behaves the same for the ordinary user as it does in Solaris. However, instead of checking only for root's UID, `mount` also checks for privileges before performing administrative tasks. The `mount(1MTSOL)` man page states that the sys_mount privilege is always required by the command in order to mount file systems and states that the command should be run with UID 0. The man page states that mount must have discretionary and MAC access both to the mount point and to the device being mounted; otherwise, MAC or DAC override privileges are required as described in `Intro(2TSOL)`. Additional override privileges may be required if any mount-time security attributes are specified for the mounted file system with the -S option. The `mount(1MTSOL)` man page also states that to succeed when mounting filesystems at any sensitivity label and when specifying any mount-time attributes, `mount` needs all of the following privileges: file_mac_read, file_dac_read, file_mac_write, file_dac_write, file_mac_search, file_dac_search. net_privaddr, proc_setsl, proc_setil, sys_mount, and sys_trans_label. Therefore, in the System Maintenance profile, the `mount` command is configured with an effective UID of 0, with required privilege sys_mount and with the other privileges that it needs to succeed in all cases.

### ▼ To Configure a New Role

1. **Define what the role's responsibilities are to be, and what commands, actions, and authorizations the role will need to do its work.**
2. **Decide whether any of the commands need privileges or other security attributes to do their work, and decide whether the role and the command can use these security attributes in a trustworthy manner.**
3. **Decide if the role needs to have a new execution profile, and if so, create one for it.**
   How to use the Profile Manager to create a new role profile is described in Chapter 8, "Managing Execution Profiles for Users and Roles." See Appendix B, "Profile Definition Tables" for lists of the existing profiles and their commands, actions, and authorizations.
4. **Create an account for the role.**
   The background for this step and the procedures to create a role account are in Chapter 5, "Using the User Manager to Configure User and Role Accounts."

# *Using the User Manager to Configure User and Role Accounts* 5

As discussed in Chapter 3, “Managing User Accounts” and in Chapter 4, “Managing Roles,” the security administrator and system administrator roles share the responsibility for entering information into the various fields of the User Manager to create and maintain accounts for users and roles.

The following two main sections in this chapter detail the types of information needed when specifying accounts for users and roles and then shows the steps for using the User Manager to add or modify user and role accounts.


| | |
|---|---|
| *Understanding the Information Entered in the User Manager Dialog Boxes* | *page 100* |
| *Setting Up or Modifying a User or Role Account* | *page 124* |


**Note –** The terms *User Manager*, *user name*, and *user ID* are retained for compatibility with earlier versions of the Solstice User Manager, even though the Trusted Solaris User Manager not only configures accounts for users but also for roles, the *user name* is actually an account name, which is a string of characters by which a user *or a role* is identified to the system, and the *user ID* is actually the account ID that can apply to either user or role accounts.

# 5

## *Understanding the Information Entered in the User Manager Dialog Boxes*

Account information for users and roles is specified by the security administrator and system administrator roles in a series of dialog boxes launched from buttons on the User Manager. See Chapter 3, “Managing User Accounts” where the division of responsibilities for configuring user accounts and the authorizations needed to access each button are described in more detail.

The buttons that launch the dialog boxes are shown in Figure 5-1.

*Figure 5-1* User Manager: Navigator

The sections listed here describe the type of information required on each of the dialog boxes and discuss choices and trade-offs among the various options you have when setting up an account.


| | |
|---|---|
| *Identity* | *page 102* |
| *User Name, User ID, Group Name(s) and Group Id(s)* | *page 102* |
| *Decisions to Make Before Starting* | *page 103* |
| *Comment* | *page 103* |
| *Decision to Make Before Starting* | *page 104* |
| *Shell* | *page 104* |
| *Using the Profile Shell To Enable Accounts* | *page 104* |
| *Using the Profile Shell To Restrict Accounts* | *page 104* |
| *Decision to Make Before Starting* | *page 105* |
| *Account Type* | *page 105* |
| *Decision to Make Before Starting* | *page 105* |
| *Password* | *page 106* |
| *Background About Creating a Password or Selecting Other Password Options* | *page 106* |
| *Decision to Make Before Starting* | *page 108* |
| *Background on the Password Duration and Warning Fields* | *page 109* |
| *Decisions to Make Before Starting* | *page 109* |
| *Background on the Account Status Menu Options* | *page 112* |
| *Decision to Make Before Starting* | *page 112* |
| *Background About Selecting a Method for Password Generation* | *page 109* |
| *Decision to Make Before Starting* | *page 111* |
| *Background About Checking NIS+ Credential Table Setup* | *page 113* |
| *Decision to Make Before Starting* | *page 113* |
| *Home* | *page 113* |
| *Why Say Yes to Automatic Creation of Home Directories?* | *page 113* |
| *Skeleton Path Considerations* | *page 114* |
| *Controlling the Use of Shell Initialization Files* | *page 114* |
| *Labels* | *page 116* |
| *Background on the Clearance and Minimum Label* | *page 116* |
| *Decisions to Make Before Starting* | *page 116* |

| | |
|---|---|
| *Background on Displaying Labels* | *page 117* |
| *Label View* | *page 117* |
| *Example of the Effects of the Label View* | *page 118* |
| *Decision to Make Before Starting* | *page 119* |
| *Background on Showing or Hiding SLs and ILs* | *page 119* |
| *Decision to Make Before Starting* | *page 119* |
| *Profiles* | *page 120* |
| *Decisions to Make Before Starting* | *page 121* |
| *Roles* | *page 122* |
| *Decisions to Make Before Starting* | *page 122* |
| *Idle* | *page 122* |
| *Decisions to Make Before Starting* | *page 123* |


## *Identity*

The system administrator enters the following information for an account in the identity dialog box:

- The name and number that identifies the user or role
- The user or role's group and (optional) supplementary groups
- A comment
- A default shell (command interpreter)
- Which type of account is being created, whether it is for a user, a non-administrative role, or an administrative role

### *User Name, User ID, Group Name(s) and Group Id(s)*

Whether the account is for a user or a role, the account's user name, user ID number, group name, and group ID number are used in these ways:

- They all are associated with each process that executes a command on behalf of the account.
- They all are associated with any files and directories created by the account.
- They all are used in DAC decisions to determine whether the user has access to files and directories created by other users based on standard UNIX file and directory permissions and access control lists (ACLs).
- For a normal user account, the user name is requested at login as part of the identification and authentication process.

- For a role account, the user name of the role account is requested when the user assumes a role through the trusted path.
- For a normal user account, the UID becomes the audit ID, which is stored in audit records that are generated by any auditable actions taken by that user. The audit ID stays the same throughout a login session, even if a user assumes a role and begins work under the role's UID. The audit ID allows administrators to trace audited actions back to individual users.

### *Decisions to Make Before Starting*

- **Decide the account's user name based on your site's convention.**
  User names cannot be longer than 8 characters.
- **Decide the account's user ID (UID) number, which is used along with the user name to identify the account on the system.**
- **Decide the primary group for the user and if the user's should belong to a secondary group or groups.**
  You need to have the group names and GID numbers available to assign to the new account.

### *Where This Information is Entered Into the User Manager*

See Step 2.a, b, c, and d on page 132.

## *Comment*

For user accounts, the Comment, which is entered in the GCOS field in the `passwd`(4) entry, contains the first and last name of the user, and perhaps the job title and work phone number. Some informal organizations allow employees to decide what goes into the comment, and sometime the entry is used for humorous purposes, such as this example for a user who writes manuals: "Roseanne Sullivan -- Manual Laborer." The text you enter for the user in the Comment: field appears in the From: line when the user sends email, and is part of the information sent about the user if someone enters `finger(1)`, among other uses:

```
From: Roseanne Sullivan -- Sun Federal Technical Writer
```

For a role account, enter whatever comment you like.

### *Decision to Make Before Starting*

♦ **Provide a comment consistent with your site's usage.**

### *Where This Information is Entered Into the User Manager*

See Step 2.a.e on page 132.

## *Shell*

The system administrator chooses a default shell for the user or role account, either Bourne, Korn, C, Profile or other. The Bourne, Korn, and C shells allow the user to execute any of the commands available on the system, limited only by the lack of any privileges or authorizations that may be needed by the command or any of its options in order to succeed. In contrast, while working in a profile shell, the account can execute only those commands that an administrative role has specified in to be the accounts set of profiles with any privileges that may be specified for each individual command.

Each time any user executes a command in the profile shell, the shell consults the profile database to find whether the command is in any of that user's execution profiles and whether any privileges are specified to be inherited by that command. If the command is in the profile database, any inheritable privileges specified in the database for the command and any authorizations in any of the account's execution profiles are made available for use while the command is executing.

### *Using the Profile Shell To Enable Accounts*

Enabling of accounts can occur because the profile shell can be used to give some user or role accounts authorizations and access to commands that may not be available to all users and to allow them to run those commands with privileges that may not be available to all users.

### *Using the Profile Shell To Restrict Accounts*

The profile shell can also be used to *restrict* users, because the profile shell, unlike other shells, may be used to limit the user to a specified set of commands.

**Note –** The profile shell is *essential* for a user when any of that user's execution profiles specify a command that needs to run with inheritable privilege, because the profile shell is the only shell that allows a process executing a command to inherit privilege. No matter what default shell has been assigned to an account, every account can invoke the profile shell on the command line in another shell.

### *Decision to Make Before Starting*

♦ **Identify the account's shell.**

### *Where This Information is Entered Into the User Manager*

See Step 2.a.f on page 132.

## *Account Type*

The system administrator role identifies whether the account is being created for a normal user, a non-administrative role or an administrative role. Choose non-administrative role or administrative role only when you are configuring a new role account as described in Chapter 8, "Managing Execution Profiles for Users and Roles."

The administrative role does its work in an administrative workspace and runs with the *Trusted Path Attribute*. Being able to work in a role workspace and to have its processes inherit the Trusted Path Attribute, also called the Trusted Path Process Flag, are the only characteristics that are different in an administrative role compared to a non-administrative role. The Trusted Path attribute allows administrative roles to do certain things, like run privilege debugging, that are not permitted to non-administrative roles.

A non-administrative role does its work in a regular user's workspace. Being able to configure non-administrative roles allows your site to assign to certain users roles that restrict the users to a controlled sets of commands, actions, and attributes to perform limited tasks, without allowing them to do all the things allowed to administrative roles. If a non-administrative role is assigned to multiple users either at the same time or one after another, each time anyone assumes the role, that person works in the same environment, shares the same home directory, and access the same files. See Chapter 4, "Managing Roles," for more about this topic.

### *Decision to Make Before Starting*

- **Choose "Normal User," if you are setting up a regular user account.** You may later specify in the Role dialog box whether this user may or may not be allowed to assume any of the *existing* roles.
- **Choose "Admin. Role" if you are setting up an account for a *new* administrative role.**
- **Choose "Non-Admin. Role," if you are setting up an account for a *new non*-administrative role.**

### *Where This Information is Entered Into the User Manager*

See Step 2.a.h on page 133.

## *Password*

The security administrator does the following in the password dialog box:

- Creates a password for the account or selects among three other options
- Specifies how long before the user *may* change the password
- Specifies how long before the user *must* change the password
- Specifies a length of time before the password expiration date to send a warning
- Chooses a password generation method to be presented to the user when the user changes the password through the Trusted Path
- Open the account, leave it closed, or specify it as "Always Open"
- Specifies whether NIS+ credentials are to be used

### *Background About Creating a Password or Selecting Other Password Options*

In most cases, when setting up an account for a user or role, the security administrator creates a password for that user or role. The security administrator passes the password on to the new user to be used at first login. It is up to the user or the site to decide how soon, if ever, the user is allowed or required to change the password for the account.

When a user is allowed to assume an administrative role, the security administrator role passes the password for the role on to that user. The role password is never used to log in directly to a role, but it is used to authenticate the user who is assuming the role through the Assume Role option in the Trusted Path menu.

Both normal users and roles use their passwords when reauthenticating themselves if the screen automatically locks after the specified idle maximum time has elapsed. The security administrator may later create another new password for a user if the user has lost the password, among other possible reasons. The security administrator role can change passwords for all other accounts except that of the security administrator role.

---

**Note –** Neither the `passwd(1)`, nor the `yppasswd(1)` or `nispasswd(1)` commands are used in the Trusted Solaris system, either by users or administrative roles. The security administrator can change an account's password through the User Manager. For each account, all password changes must be done through the Change Password option on the Trusted Path Menu after the account logs in and is authenticated by the system. Normal users change passwords from the Trusted Path menu in a user workspace. Users in roles change the role passwords from the Trusted Path menu in a role workspace.

---

When either of the Type in or Choose from list are selected from the Password menu, the administrative role is immediately prompted to create a password for the user by the method selected.

---

**Note –** Only the "Type in" and "Choose from List" options on the Password menu are recommended because those are the only two options consistent with Trusted Solaris security policy for user identification, authentication, and accountability. Any of the other options should be used only in specialized circumstances where a knowledgeable security administrator ascertains that

the option is both needed and allowable within the site's security policy—keeping in mind that using it makes the system both more vulnerable and harder to maintain.

*Table 5-1* Password Creation Options, Descriptions and Recommendations

| Password Creation Option | Description |
|---|---|
| Cleared until first login | Account is prompted for a password at first log in. Selecting this option requires accounts to choose a password before they have been identified and authenticated. If a new user account is created with this option, anyone could log in using the username and supply any password. Therefore, this option is not recommended for sites that want to observe the requirement that password changes should be only done by authenticated users. |
| Account is locked* | When this option is selected, the account is locked with an invalid password and can be unlocked by assigning a new password. With this option, the account can own files but cannot log in until the account is unlocked. |
| No password -- setuid only | This option sets up a specialized account that can own files but that can never be logged into, and it is never selected for user or role accounts. Accounts are set up with this option only for use by programs that use `setuid` to change their identity to the ID of the account so that they can access files owned by the account. Existing accounts of this type on the default system include `lp` and `uucp`. Being able to create no-password accounts helps reduce the risk of intruders deducing the passwords for these accounts for the purpose of breaking into the system. |
| Type in | Selecting this option brings up the Password Add dialog box for the administrative role to enter the account's password. |
| Choose from list | Selecting this option brings up the Password generation dialog box with a list of automatically generated passwords from which the administrative role chooses the account's password. |

* This option is not the same as the method used in Trusted Solaris 1.x for locking an account until the security attributes had been specified. In Trusted Solaris 2.x, the "Closed" option in the Status menus used instead. See Table 5-3 on page 112.

### *Decision to Make Before Starting*

- **Decide whether to create a password for the account by typing it in or choosing from an automatically-generated list.**

### *Where This Information is Entered Into the User Manager*

See Step 3.a. i, ii, and iii on page 137.

## *Background on the Password Duration and Warning Fields*

The password change options limit how long intruders could potentially access the system if they were able to guess or steal passwords. Establishing a minimum length of time to elapse before change stops users who have just been given new passwords from reverting immediately to their old passwords.

### *Decisions to Make Before Starting*

- **Decide how long before the user *may* change the password,**
- **Decide how long before the user *must* change the password, which is specified by either:**
  - Entering a number of days, weeks or months after "Max Change" or
  - Choosing an Expiration Date from the Days, Months, and Years menus.
- **Decide the number of days, weeks, or months or a date before the password is to expire for a warning to be sent to the user.**

### *Where This Information is Entered Into the User Manager*

See Step 3.c. i on page 137, ii, iii, iv, and Step 3.d on page 138.

## *Background About Selecting a Method for Password Generation*

The method of password generation selected in the Change by menu determines whether the manual or automatic password generator dialog box is presented whenever the account changes the password through the Trusted Path.

The rules for passwords in Table 5-2 apply to manually-generated passwords that the account types in when modifying the password through the trusted path.

*Table 5-2* Password Rules for Manually Created Passwords *(1 of 2)*

| **Rules** |
|---|
| The maximum number of characters is eight. |
| The minimum number of characters is six. |
| The password must contain at least two alphabetic characters. |

*Table 5-2* Password Rules for Manually Created Passwords *(2 of 2)*

| Rules |
| --- |
| The password must contain at least one numeric or special character. |
| The password must differ from the user's login name and any reverse or circular shift of that login name. (For this comparison, upper case letters and lower case letters are considered to be equal.) |
| A new password must have at least three characters different from the old. (For this comparison, upper case letters and lower case letters are considered to be equal.) |

**Note –** The rules for passwords created using the User Manager, and the passwords created by the password generation software are slightly different from those shown in  on page 109 in that, for example, the automatically-generated passwords do not have numbers, and the User Manager-generated passwords have other requirements. Although generated passwords and the User Manager-created passwords do not fully conform to the rules for manually-generated passwords, these passwords are also accepted by the system.

The values for passwords generated through the Trusted Path are set in the pluggable authentication module (PAM) values in the header file `pam_headers.h`, shown in Figure 5-2 on page 111.

**Note –** The MINWEEKS, MAXWEEKS, and WARNWEEKS #defines in the PAM header file are set to -1, which is not a valid number, which allows these values to be set through the User Manager for each user.

```
/*
 * Miscellaneous constants
*/
#define NUMCP          13     /* number of characters for valid password */
#define PAM_TRY_AGAIN   1
#define MAX_INSIST      3   /* 3 chances to enter new passwd */
#define MINWEEKS       -1   /* minimum weeks before next password change */
#define MAXWEEKS     -1   /* maximum weeks before password change */
#define WARNWEEKS    -1   /* number weeks before password expires */
                          /* to warn the user */
#define MINLENGTH     6   /* minimum length for passwords */
#define ROOTUID       0
```

*Figure 5-2* System-wide PAM Settings

### *Decision to Make Before Starting*

Decide whether the account can pick its own password or if it must pick one from an automatically-generated list.

### *Where This Information is Entered Into the User Manager*

See Step 3.e on page 139.

## Background on the Account Status Menu Options

The Status menu options are new to the Trusted Solaris version of the base Solstice User Manager. See Table 5-3 for descriptions and recommendations for each of the status options.

**Note –** The user's account is closed after three failed login attempts. This helps protect against penetration of the system by repeated attempts to guess passwords. Trusted accounts can be exempt from this restriction if the Always Open option is set for the user.

*Table 5-3* Status Menu Options, Descriptions, and Recommendations

| Status Option | Description and Recommendation |
|---|---|
| Closed | In effect until the account has been completely specified. Unlike account locking, which is specified in the Password menu near the top of the Password dialog box (see Table 5-1 on page 108). this option does not affect the account's password. After three (3) failed login attempts, the status of the account is automatically changed to closed. [The word "locked" is entered into the "lock" field in the user's entry in the `tsoluser`(4) password file.] The security administrator role then must use the User Manager to reset this value to Open. |
| Open | Use after the account has been completely specified or, using discretion, reset to this value after the account has been locked. The security administrator role chooses this option after setting up the security-relevant characteristics of the user account., or if the account has been locked after 3 failed login attempts, resets this value after verifying the t This value is not accepted by the User Manager until the account is fully specified. |
| Always Open | Use only when your site's security permit an account to stay open even if the limit on failed logins is reached. This option may be used at site's where the need for protection against the denial of service (which would occur if a malicious user attempted multiple logins with the attention of closing accounts) is greater than the need to limit attempts to penetrate the system by repeated attempts to guess passwords. |

### *Decision to Make Before Starting*

♦ **Decide the account status.**

### *Where This Information is Entered Into the User Manager*

See Step 3.f on page 139.

### *Background About Checking NIS+ Credential Table Setup*

Clicking the toggle button next to the Cred. Table Setup field, adds the NIS+ principal's public and private keys to the `cred` table, which establishes the account as a NIS+ principal and adds the accounts password to the NIS+ databases. (See Chapter 5, "Administering NIS+ Credentials," of the *NIS+ and FNS Administration Guide - Solaris 2.5.)*

#### *Decision to Make Before Starting*

- **Decide whether to check the Cred. Table Setup box for NIS+ credential setup.**

#### *Where This Information is Entered Into the User Manager*

See Step 3.g on page 140.

## *Home*

The system administrator sets the following in the Home dialog box:

- Whether or not the home directory is automatically created
- The home directory's pathname
- The name of the server for the home directory
- The skeleton path for the shell initialization files
- The desired home directory DAC permissions
- The name of the account's mail server

---

**Note –** Because you are prompted to specify a server for the account's home directory, the server for the account's home directory must be configured before you create the account.

---

### *Why Say Yes to Automatic Creation of Home Directories?*

If the system administrator does not tell the system to automatically create the home directory, then the system administrator must set up the home directory manually as an MLD before the user can log in. Because the system can

automatically copy start-up files to the initial SLD within the multilevel home directory the first time an account logs in, it is advisable to have the system do the set up.

### *Skeleton Path Considerations*

When you are prompted to specify a skeleton path for shell startup files, if you specify `/etc/skel` as the skeleton path location without making any changes, the supplied `local.cshrc`, `local.login` and `local.profile` start-up files for all the shells are automatically copied into the account's home directory. The account is responsible for renaming the appropriate copied file and removing the files that do not apply to her or his shell. For example, if the user's login shell is the C shell, the user would remove the `local.profile` and would rename and make any desired modifications to the `local.cshrc` file and local.`login` files.

---

**Note –** Without administrative intervention, neither the `.login` or `.profile` are looked at when the window system comes up unless the account's default shell is `pfsh`. Also, files in a skeleton directory are copied only into the first home directory SLD created at the account's initial sensitivity label, and the administrator or user must ensure that startup files are copied to all subsequent SLDs created at other sensitivity labels for the account. The needed background and procedures related to startup files are described in "Managing Startup Files in a Trusted Solaris System" on page 65 in Chapter 3, "Managing User Accounts."

---

### *Controlling the Use of Shell Initialization Files*

*When a user's login shell is specified as either Bourne, Korn or C shell*, keep in mind that, by default, shell initialization files are not looked at during login because a variable in a `*.dtprofile` file is commented out. Other comments in the `*.dtprofile` files encourage the administrator or individual users to modify the initialization files before removing the comment if they wish the appropriate initialization file for a shell to be sourced, and to make sure that the initialization file does not do

anything that requires a terminal emulator or user interaction while the window system is coming up.

A personal $HOME/`.dtprofile` file could be modified in such a way that it would allow the user to launch commands while the window system is coming up and before the user's profiles are in effect. To support the special purposes of the profile shell, `pfsh`, it is important that users not be allowed to specify any commands that run before the user's shell is in effect. So *when a user's login shell is specified as the profile shell*, the `.profile` file *is* looked at during login, but the profile-shell-user's $HOME/`.dtprofile` file is never consulted.

For more details on these topics, see "To Make .login and .profile Looked at During Login for Bourne, C and Korn Shell Users" on page 83 and see "Managing Startup Files in a Trusted Solaris System" on page 65 in Chapter 3, "Managing User Accounts."

### *Decisions to Make Before Starting*

- **Decide the account's home directory server.**
- **Decide whether to allow the automatic creation of a multilevel home directory (MLD) for the account.**
- **Decide what to put in `/etc/skel`, whether to let the users do their own modifications to shell initialization files or whether to provide your own administratively-controlled versions. If the latter, you must also do the set up described in "To Separate the Shell Initialization Files for Each Shell" on page 83 in Chapter 3, "Managing User Accounts.**
- **If a Trusted Solaris administrative role has created a shell-specific subdirectory in `/etc/skel` for each of the shells, prepare to enter the correct `skel`*X* subdirectory name into the Skeleton path field in the User Account Manager, based on the user's default shell.**
- **Decide the default permission bits for files the user creates (for setting the `umask`).**
- **Decide the mail server for the account.**

### *Where This Information is Entered Into the User Manager*

See Step 4.a on page 141, c, d, e, f, g, and h on page 143.

# 5

## Labels

The system administrator specifies the following in the Labels dialog box:

- The clearance
- The minimum SL
- Whether the account is allowed to view administrative labels or should be shown alternate labels within the user accreditation range
- Whether the user can view SLs
- Whether the user can view ILs

### Background on the Clearance and Minimum Label

The clearance defines top of the range of sensitivity labels at which the account can work. Administrative role accounts have a clearance of ADMIN_HIGH.

The account's minimum sensitivity label defines the lowest sensitivity label at which the account can work. Administrative role accounts have a minimum sensitivity label of ADMIN_LOW. The minimum sensitivity label is the sensitivity label of the first workspace that comes up for the account after the first login.

During the first session or during any subsequent session, an account can change the sensitivity label of any workspace and can specify an alternate workspace so that any workspace at any sensitivity label within the user's personal label range may be first workspace that comes up during subsequent sessions.

---

**Note –** Both the account's clearance and minimum sensitivity label must be dominated by the highest sensitivity label defined in the user accreditation range and must dominate the minimum clearance defined in the user accreditation range in the `label_encodings(4TSOL)` file.

---

#### *Decisions to Make Before Starting*

- **Decide a clearance for the account that dominates the minimum clearance set in the `label_encodings` file.**
- **Decide the account's minimum sensitivity label, which is the minimum sensitivity label at which the account is permitted to work.**

### *Where This Information is Entered Into the User Manager*

See Step 5 on page 143, Step 5.a on page 144, Step 5.b on page 146.

## *Background on Displaying Labels*

The system administrator specifies how labels are displayed for the account in the lower part of the Labels dialog box. In the following discussion, the term CMW label is used to refer to the IL and the SL when they are shown together in the standard format, with the long name of the information label and any words and the short name of the sensitivity label and of any words.

```
INFORMATION LABEL [SL]
```

### *Label View*

The label view allows the security administrator to determine whether the ASCII names for administrative labels are ever displayed for the account, because at some sites the names of administrative labels are considered to be classified information. ADMIN_HIGH and ADMIN_LOW are the default ASCII names for the administrative labels, but because the security administrator role is able to specify alternate names for administrative labels in the `label_encodings(4TSOL)` file, your site could have other names assigned.[1] The administrative labels appear in sensitivity labels, information labels, and clearances.

A system-wide default label view is set in the `label_encodings` file. During installation, the default setting of INTERNAL is either accepted or changed by the site's security administrator. When each user or role's account is being set up, the security administrator chooses one of the following for the account:

- Internal
- External
- Sys Default

### *Internal View*

The INTERNAL view allows the account to see the *names* of the administrative labels, which are either the strings "ADMIN_HIGH" and "ADMIN_LOW" or their administratively set names.

1. See *Trusted Solaris Label Administration* for how the security administrator role specified alternative names.

### *External View*

Accounts with the External option are not exposed to the ASCII names of the administrative labels. If the label view for an account is set to EXTERNAL, the *minimum* valid label of the same type in the User Accreditation Range is shown instead of the ADMIN_LOW label or its site-specified equivalent. Also, when the account's label view is EXTERNAL, the *maximum* valid label of the same type in the User Accreditation Range is shown instead of the ADMIN_HIGH label or its site-specific equivalent.

**Note –** It is important to realize that the binary label always remains the same when the EXTERNAL view is set. The only difference is that the label is given an alternative name when it is displayed to hide its real name.

### *Sys Default*

If the Sys Default option is selected for an account, whatever value is specified in the `label_encodings(4TSOL)` file for the "DEFAULT LABEL VIEW" key word (EXTERNAL or INTERNAL) in the file applies to the account.

### *Example of the Effects of the Label View*

Here is an example of how the default label view affects what the user sees. Remember that the IL of a newly opened file is always ADMIN_LOW because it is empty, while its SL is the label of the process that created it. So, if a user begins to edit a file in a Text Editor at TOP SECRET at a site where the default administrative label names are being used, the CMW label is:

```
ADMIN_LOW [TS]
```

In the example, the lowest valid IL in the user accreditation range is CONFIDENTIAL. If the account has the INTERNAL label view set, the CMW label displays with the administrative label ADMIN_LOW in the trusted frame of the Text Editor as: `ADMIN_LOW [TS]`.

If the account has the EXTERNAL label view set, the CMW label displays with the label UNCLASSIFIED replacing the administrative label, as: `UNCLASSIFIED[TS]`.

**Note –** Ultimately, whether the IL portion and the SL portion of administrative labels or their substitutes display, or whether the administrative labels display at all, is determined by the values set in the SL and IL display menus below the label view menu.

### *Decision to Make Before Starting*

- **Decide whether the user is allowed to see the names of administrative labels or if the user will see the minimum valid label in the User Accreditation Range instead of the ADMIN_LOW label and see the maximum valid label in the User Accreditation Range instead of the ADMIN_HIGH label.**

### *Where This Information is Entered Into the User Manager*

See Step 5.c on page 148.

## *Background on Showing or Hiding SLs and ILs*

When the complete CMW label, the SL, or the IL are displayed, they appear in the following locations, among others:

- The trusted path indicator at the bottom of the screen,
- At the top of window frames, and
- In the title bar on window icons.

When the IL and SL are displayed together in the form of a CMW label, SLs display in short form inside square brackets ([]). When the SL displays alone, it displays in long form within the square brackets.

### *Decision to Make Before Starting*

- **Decide if the account can view the complete CMW label, the Sensitivity Label alone, or the Information Label alone.**

### *Where This Information is Entered Into the User Manager*

Step 5.d, and e on page 148.

# 5

## *Profiles*

The security administrator role assigns execution profiles to accounts. The available execution profiles are displayed in a scrolling list in the Profiles dialog box, along with brief descriptions. See Appendix B, “Profile Definition Tables” for tables that list the actions, authorization, and commands defined for each profile.

---

**Note –** The default administrative roles provided with Trusted Solaris have the needed execution profiles already assigned. If you create a new role, you may create a new profile for that role; if so, create the new profile before you configure the role account, so that you can assign the profile to the role’s account through the User Manager.

---

If you are configuring a user account and plan to assign a role to that user, *you do not need to and should not assign a corresponding role profile to the user’s account*. It’s important to realize that the role accounts have their own execution profiles, which come into effect only when the user assumes the role and in effect, changes his or her identity to that of the role.

---

**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. To begin with, role profiles you assigned to the user account would be in effect for the user *when the user has not assumed a role*, which in most cases should not be allowed to happen. What’s more, some things that seem like they should work would not: 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.

---

The *order of profiles* is also important because when the account invokes a command or action, the profile mechanism uses the command or action the *first* time its name is found in any of the profiles in the account’s profile set—with whatever attributes have been defined for the command or action in the profile where it is found. This is similar to how the system uses the first instance of a command that it finds in any directory in the user’s PATH variable: for example, if you invoke the `ps(1)` command by entering `ps` on the command line, either `/usr/bin/ps` or `/usr/ucb/ps` will run depending on how your PATH is set up, and each version has its own set of options.

Similarly, if the user executes a command or action that is defined with differing attributes in more than one execution profile (for example, with differing privileges or with `setuid` turned on in one but not the other), the *command or action runs as it is defined in the first profile in which it is found in the profile list*.

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

Also, it is possible to take advantage of the order of profiles to allow certain accounts to run certain commands and actions with privileges or with an effective UID or GID or at a certain sensitivity label and run all other commands and actions with none of these special security attributes. Do this by assigning a set of profiles that explicitly names any commands or actions that the account needs to run with the specific security attributes and then, at the end of the list of profiles, give that account the All profile that permits the account to run any other command or action in the system without privilege. In this way, the account can use the explicitly named and configured commands and actions with the specified attributes and use all other commands without any special attributes.

For more about execution profiles see “Review of Terms” on page 194 in Chapter 8, “Managing Execution Profiles for Users and Roles.”

### *Decisions to Make Before Starting*

- **Decide on at least one execution profile to assign to the account.**
- **Decide the order in which execution profiles should be listed**

### *Where This Information is Entered Into the User Manager*

Step 6.a and b on page 150, and Step 6.c and d on page 151.

## Roles

The security administrator role assigns roles to normal user accounts. The available roles are displayed in a scrolling list in the Roles dialog box, along with brief descriptions.

Role accounts cannot assume other roles, and the User Manager does not allow access to the Role dialog box when a role account is being configured. However, you may decide to assign multiple roles to a single user if it is consistent with your site's security policy.

### *Decisions to Make Before Starting*

♦ **Decide which roles, if any, the account is allowed to assume.**

### *Where This Information is Entered Into the User Manager*

Step 7 on page 151.

## Idle

The security administrator role specifies whether an action is taken if the workstation remains idle for an amount of time that is specifiable by the role. The two action choices are to log out the user or lock the screen.

- Lock screen

  Locks the screen after the specified period of idleness has passed. The account must then supply a password to regain access to the session. Moving the mouse or pressing a key causes the dialog box shown in Figure 5-3 on page 123 to display so that the user can enter the password.

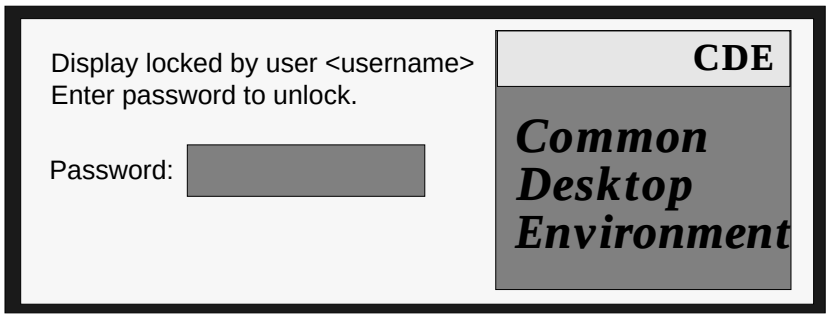



*Figure 5-3* Lock-out Password Dialog Box

- Logout –

  Logs the user out of the system entirely when the specified period of idleness has passed. The user must log in again to regain access.

The security administrator role also chooses from the Idle Time menu an amount of time before either action, either 1, 2, 3, 4, 5, 10, 15, 30, 60, or 120 minutes or Forever. The Forever menu option essentially prevents any action from being taken when the workstation is idle.

### *Decisions to Make Before Starting*

- Decide on the consequences, if any, of a workstation sitting idle when a user is logged on but not active, if the screen should be locked, or the user's session should be terminated and the user logged out.
- Decide what length of time must pass without activity on the workstation before the specified action will be taken.

### *Where This Information is Entered Into the User Manager*

Step 8 on page 153.

# 5

## *Setting Up or Modifying a User or Role Account*

The following procedures show how to use the various options on the User Manager to specify account information.


| | |
|---|---|
| *To Launch the User Manager* | *page 124* |
| *To Load a List of User and Role Accounts Using the Load Dialog Box* | *page 125* |
| *To Load Users or Exit (optional)* | *page 127* |
| *To Find or Sort Accounts* | *page 128* |
| *To Add, Modify or Delete Accounts* | *page 129* |


### ▼ To Launch the User Manager

1. **Click the Application Manager icon in the front panel.**
   **(See Figure 5-4 on page 125 for this and the following step.)**
2. **Click the Solstice_Apps folder icon in the Application Manager.**
3. **Click the User Manager icon.**
   The User Manager: Load dialog box displays as shown in Figure 5-5 on page 126 and the main User Manager window displays with no users.

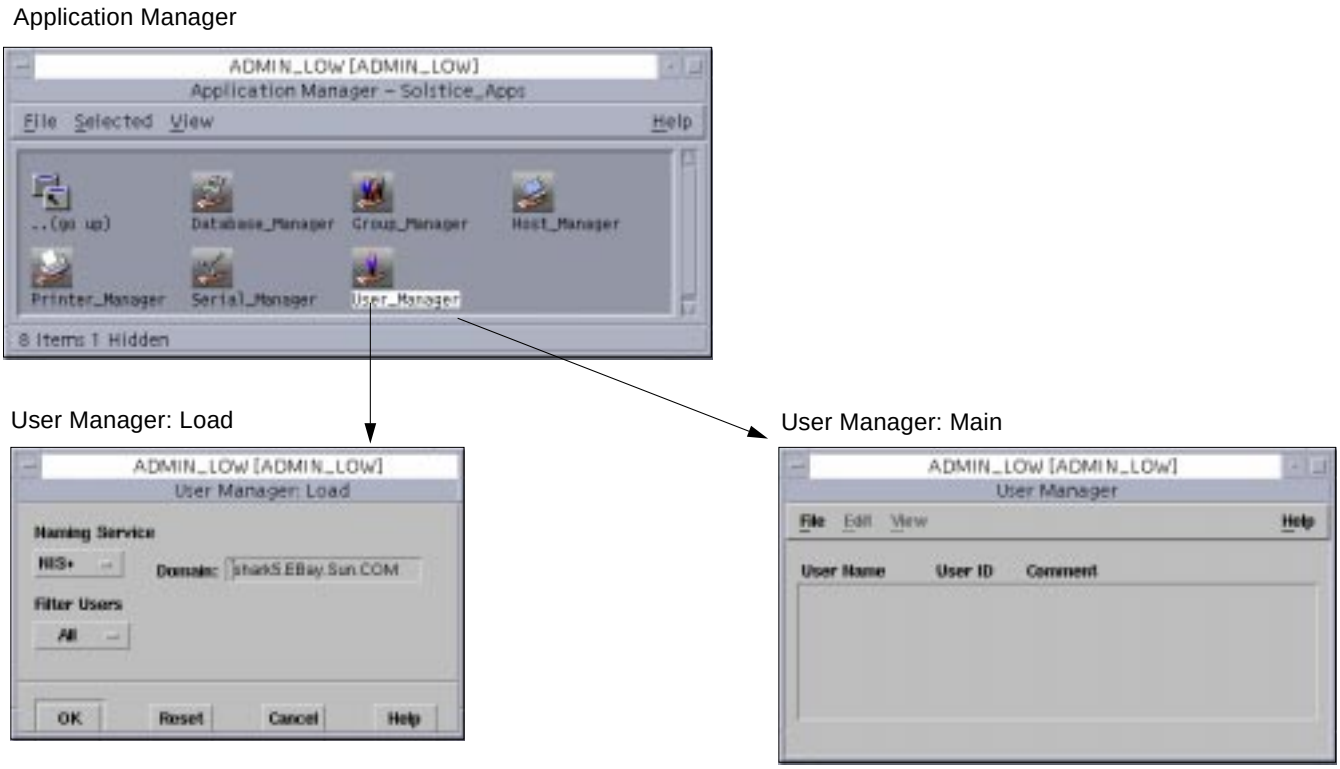



*Figure 5-4* Launching the User Manager

### ▼ To Load a List of User and Role Accounts Using the Load Dialog Box

1. **From the Naming Service menu, choose the NIS+ option.**

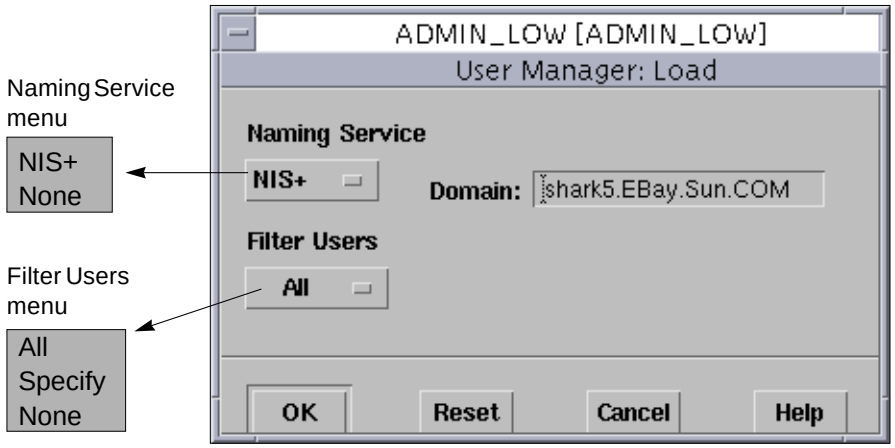



*Figure 5-5* User Manager: Load Dialog Box with Filter Users Menu

**Note –** NIS+ is the recommended option in the Trusted Solaris system, because it supports the centralized administration of all the workstations and servers together in one a single distributed system, which is important for both user accountability and trusted administration. The None option should only be used in specialized circumstances where a knowledgeable security administrator decides that local accounts are both needed and allowable within the site's security policy— even though they can make the system both harder to protect and harder to maintain.

2. **Accept the domain name shown or enter another domain name in the text field next to Domain.**

3. **Choose an option from the Filter Users menu.**

   a. **Choose All to display a** list of names of already-configured accounts along with their associated user IDs and comments**.**

   b. **Choose Specify to specify a user or set of users.**
   Next to the Specify option, do either i. or ii.

      i. **Enter a specific username.**

      ii. **Enter a regular expression to filter the usernames.**

   c. **Choose None to cause the main User Manager: Load dialog box to be displayed with no account names.**

4. **Click OK.**

## ▼ To Load Users or Exit (optional)

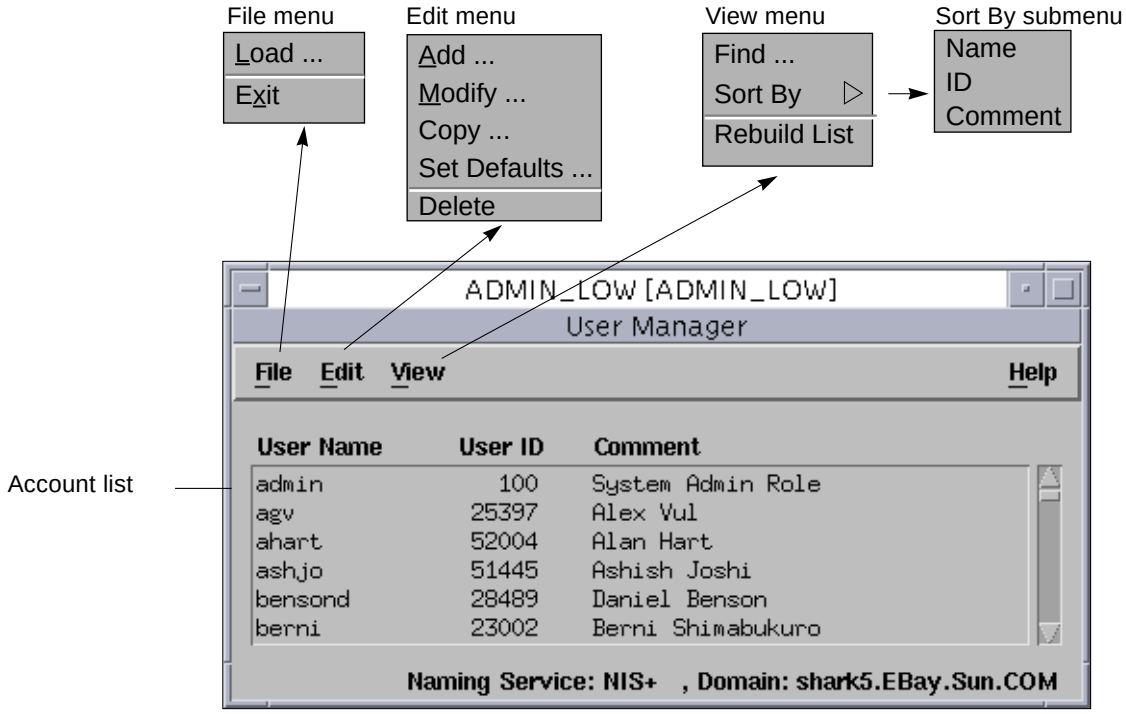



*Figure 5-6* User Manager: Main Window and Menus

1. **Choose the Load option from the File menu to bring up the Load dialog box (see Figure 5-6).**

   See "To Load a List of User and Role Accounts Using the Load Dialog Box" on page 125 and Figure 5-5 on page 126 for how to use the Load dialog box

2. **Choose the Exit option to exit from all User Manager windows.**

## ▼ To Find or Sort Accounts

1. **Choose an option from the View menu.**
   The View menu options are shown in Figure 5-7.

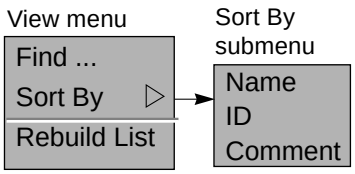



*Figure 5-7* View Menu with Sort By Submenu

   a. **Choose Find to enter a target string (containing zero or more wildcards) for locating a user in the list.**

      i. **Enter either a username, user ID, or comment.**
      See Figure 5-8. You can only specify one type of target at a time.

      ii. **Click Apply or OK.**
      OK highlights the next matching user on the User Manager list, if any, and then the User Manager: Find dialog box disappears. Apply highlights the next match, if any, and the Find dialog box continues to display. If there are multiple matches, click Apply again on the User Manager: Find dialog box to go to the next match.

*Figure 5-8* User Manager: Find Dialog Box

   b. **Choose Sort By to display a submenu and select one of the three fields to sort by: Name, ID, or Comment (see Figure 5-7).**

   c. **Choose Rebuild List to display the current list of accounts with any additions, deletions or modifications made since the list was last displayed.**

## ▼ To Add, Modify or Delete Accounts

**Note –** As stated in Chapter 3, “Managing User Accounts,” the responsibilities for setting up accounts are divided between the security administrator role, which handles the security aspects of the user’s record, and the system administrator role, which handles the general aspects. Another security feature is that a role cannot modify its own account.

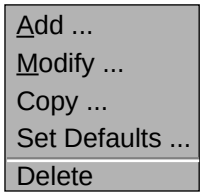



*Figure 5-9* User Manager Edit

1. **Choose Add, Modify, Copy, Set Defaults, or Delete from the User Manager Edit menu shown in Figure 5-9.**
   Selecting any of the options from the Edit menu displays one of the Navigator dialog boxes.

   The Navigator (Add) dialog box is shown in Figure 5-10. The buttons in the Navigator give access to additional dialog boxes for entering the named types of account information.

**Note –** If any buttons are grayed out when an administrative role brings up the User Manager, this means that the authorization needed for that button is not in the role’s execution profile. The system administrator role first fills out the fields and specifies the information for which the role is authorized, and the security administrator role then specifies the rest of the information and sets the account status to Open.

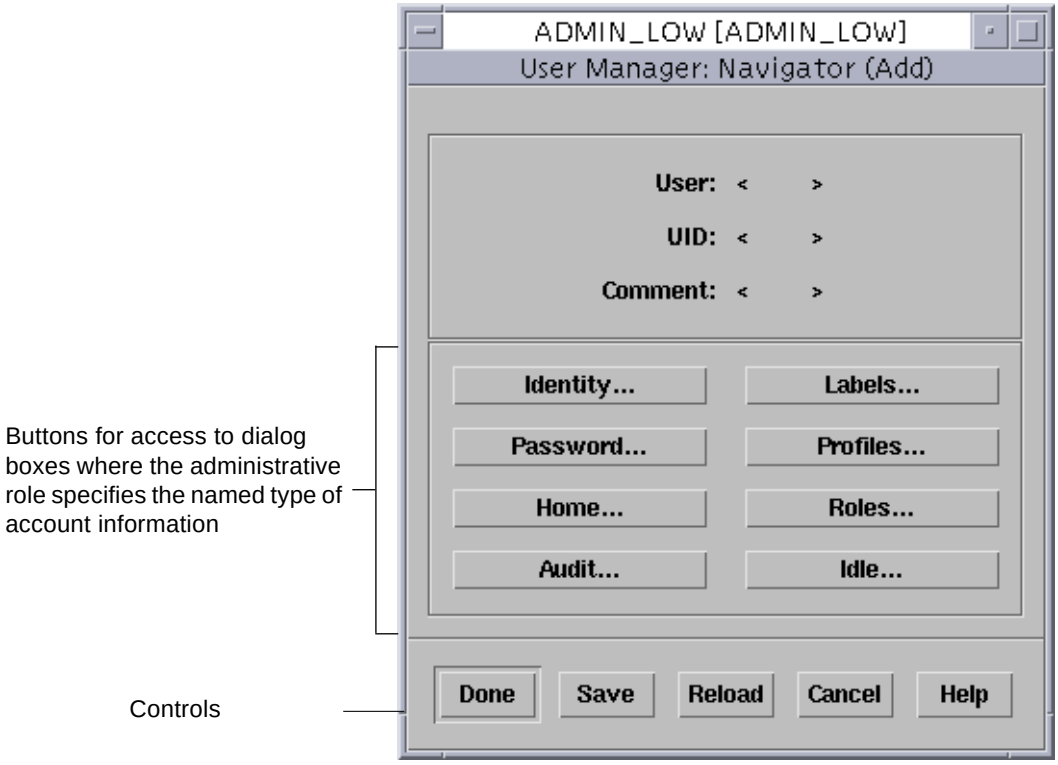



*Figure 5-10* User Manager: Navigator

**Note –** On all the user manager dialog boxes, clicking the Apply button saves the data and leaves the dialog box displayed, while clicking the OK button registers the data and closes the dialog box.

**Note –** The Audit button is not enabled in the 2.5 release of the Trusted Solaris operating system. If you click on it, it does not respond. See the *Trusted Solaris Audit Administration* manual for how to set up auditing for individual users.

**d. To set defaults to apply to all new accounts, choose Set Defaults, and then click the button that allows you to update the desired type of information (optional).**
The defaults you can set on each of the dialog boxes are shown in Table 5-4. The defaults you cannot set are grayed out.

*Table 5-4* Defaults You Can Set on the User Manager Dialog Boxes

| Dialog Box | Settable Defaults | For Instructions, Go to: |
|---|---|---|
| Identity | Primary Group | Step 2 on page 131 |
| | Secondary Group | |
| | Log in Shell | |
| | User Type | |
| Password | All | Step 3 on page 133 |
| Home | All except Path | Step 4 on page 141 |
| Labels | All | Step 5 on page 143 |
| Profiles | All | Step 6 on page 149 |
| Roles | All | Step 7 on page 151 |
| Idle | All | Step 8 on page 153 |

**e. When adding a new account, choose Add or Copy, and then go to Step 2.**
Copy lets you create a new account by editing the fields already specified for an existing account. This option allows you to keep the entries you want from another user account while changing only the information that is different for the new account.

**f. When modifying an existing account, choose Modify, and then go to Step 2.**

**g. When removing an account, choose Delete, and then go to Step 9 on page 155.**

**2. Enter a new account's identity Information, or change it for an existing account if desired.**
See "Identity" on page 102, if necessary, to review the guidelines on what to specify in the Identity dialog box.

**a. Click on the Identity button in the User Manager Navigator.**
The Identity dialog box displays (see Figure 5-11).

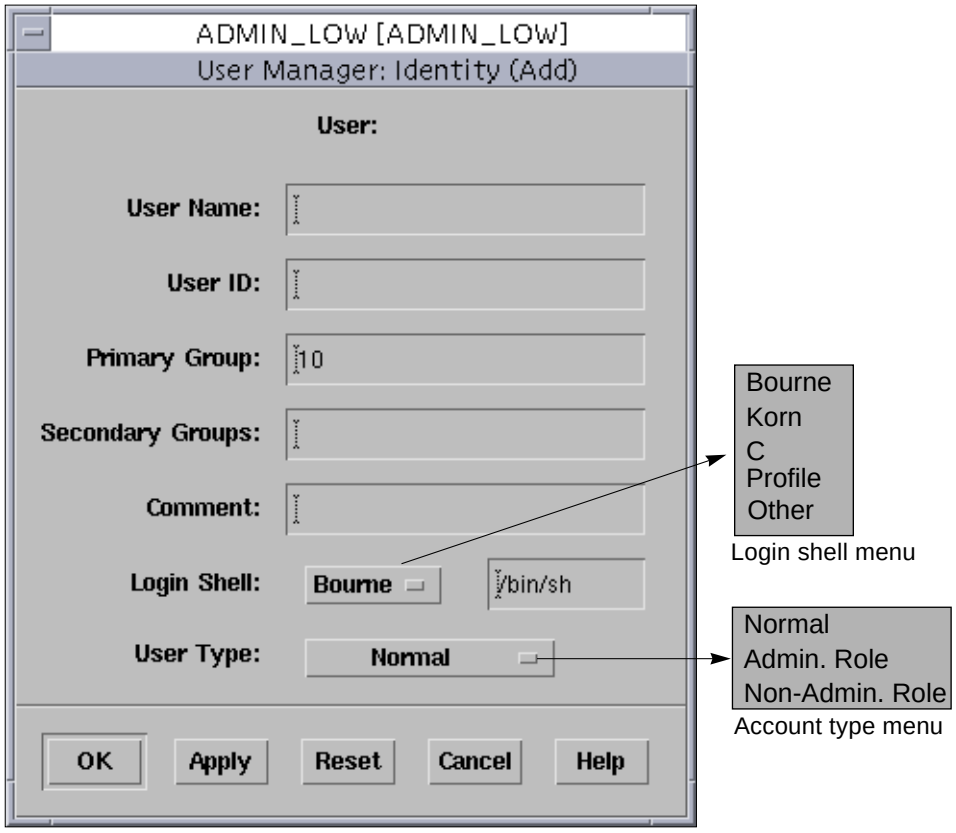



*Figure 5-11* User Manager: Identity Add Dialog Box

**b. Enter the name that identifies the user or role.**
The name cannot be longer than eight characters.

**c. Enter the numeric ID for the user or role.**

**d. Enter the user or role's group and (optional) supplementary groups.**

**e. Enter a comment.**

**f. Choose a default shell from the Login Shell menu: Profile, Bourne, Korn, C, or another type that you specify.**

**g. If you select other, enter the pathname of the shell.**

---

**Note –** The profile shell should *always* be selected as the login shell for role accounts. Normal users may be configured to have the profile shell if you want use it to take advantage of its enabling or restricting capabilities.

---

**h. Choose the type of account from the User Type menu.**

Choose either normal user, administrative role, or non-administrative role.

**i. Store your changes by clicking Apply or OK at the bottom of the Identity dialog box. (See Figure 5-12).**

Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

*Figure 5-12* Controls on the User Manager: Identity Dialog Box

**j. If you are done entering information for a new account or to update an existing account, go to Step 9 on page 155.**

**3. Enter a new account's password information, or change it for an existing account, if desired.**

See "Password" on page 106, if necessary, to review the guidelines on what to specify in the Password dialog box.

**a. Click the Password button in the User Manager Navigator.**

The Password dialog box displays (see Figure 5-13).

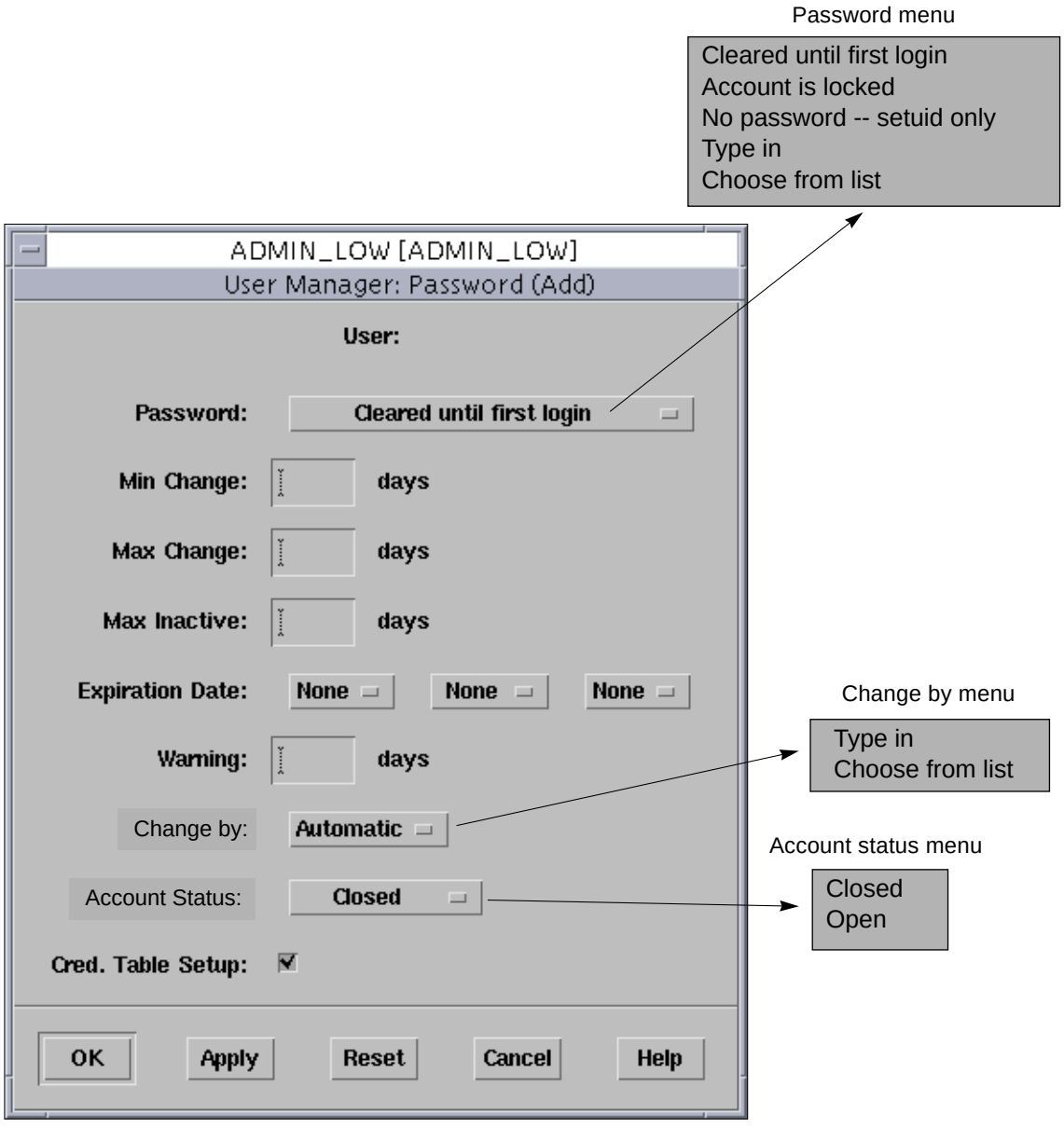



*Figure 5-13* User Manager: Password Dialog Box

**b. Choose either** `Type in` **or** `Choose from list` **from the Password menu (see Figure 5-14).**

*Figure 5-14* Password Dialog Box: Password Menu

If you choose Type in, the dialog box shown in Figure 5-15 displays. If you choose the Choose from list option, the dialog box in Figure 5-17 on page 138 displays.

**i. For a typed in password, enter the password in the Enter Password text field, press the Tab key to go to the Verify Password text field, and enter the password again (see Figure 5-15).**

*Figure 5-15* User Manager: Set Password

**ii. To choose a password from a generated list in the Password Generator dialog box, click the circle to the left of the desired password, and then enter the password you selected into the text entry field below to confirm (see Figure 5-16).**

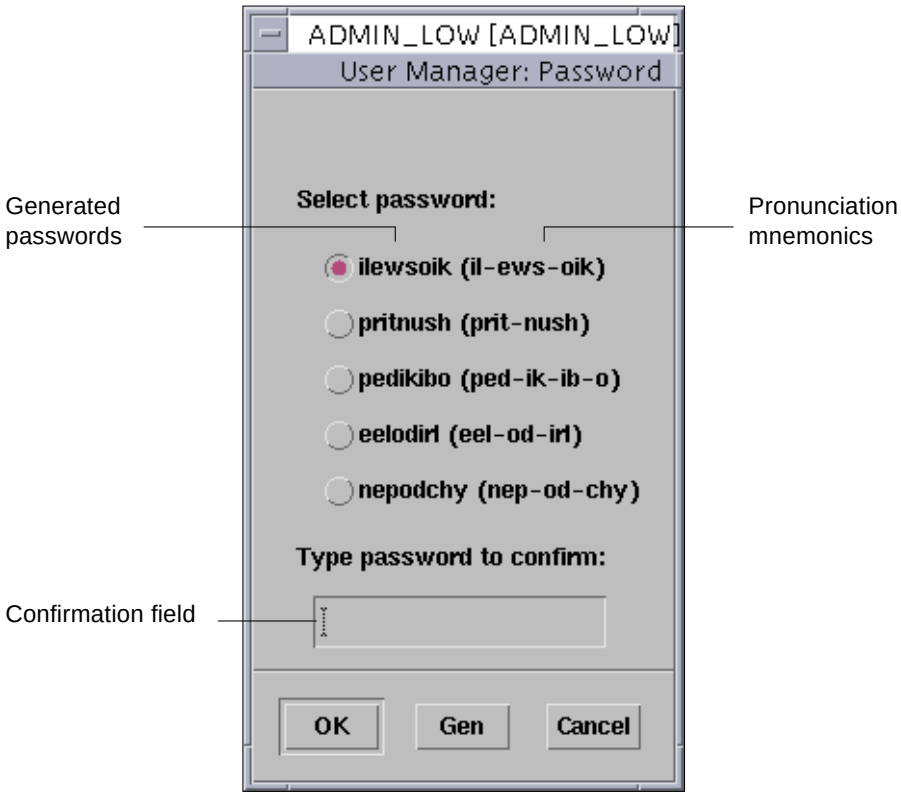



*Figure 5-16* Password Generator Dialog Box

The Password Generator dialog box gives a choice of five system-generated passwords. The pronunciation mnemonic shown in parentheses to the right of each password divides the password into syllables to make it easier to remember. If you do not find a generated password that you like, go to Step iii on page 137

iii. **If you do not find a generated password that you like, click Gen to get five new generated passwords to choose among, and select one, as described in Step ii on page 136.**

iv. **Either click OK or press Enter to store the password.**

c. **Specify the duration of password validity (see Figure 5-17 on page 138) (optional).**

i. **In the Min Change field, enter a minimum number of days that must elapse after a password change before the user may change the password on the account again.**

ii. **To have the password expire by a certain number of days after its last change, put that number of days in the Max Change text entry field.**

iii. **In the Max Inactive field, enter a maximum number of days that an account can be inactive before the user account is closed.**

iv. **To have the password expire on a certain date, enter the date in the Expiration Date field.**

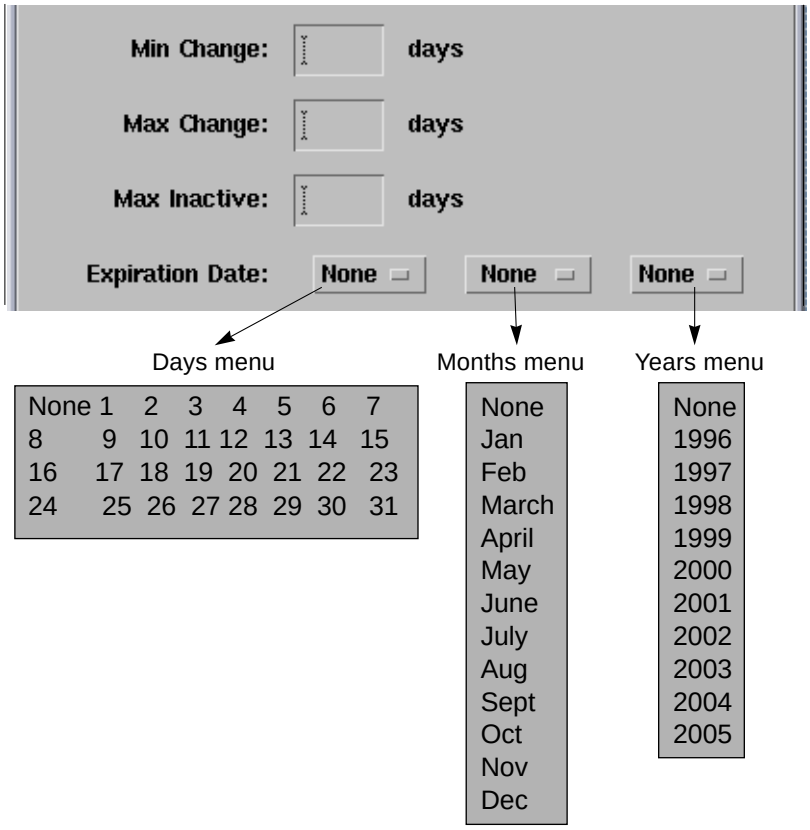



*Figure 5-17* Password Dialog Box: Password Duration and Expiration Date Fields

**d. If a password validity duration or expiration date is set in Step c, type into the Warning text entry field a number of days before the password expires for a warning to be sent (see Figure 5-18 on page 138.)**

*Figure 5-18* Password Dialog Box: Warning Field

**Note –** Entering meaningless combinations such as 1-none-2000 generates errors.

**e. Set the password generation type for the user**

The Generation menu in the Password dialog box lets you specify whether the user or the system is to supply the password (see Figure 5-19).

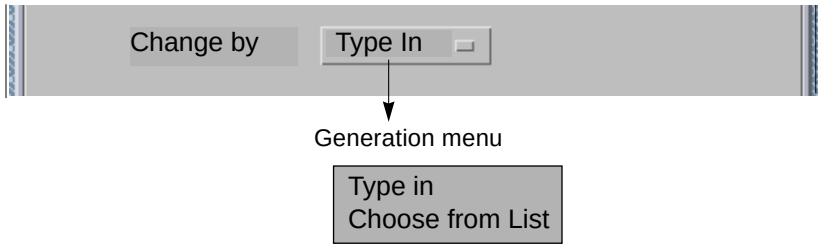



*Figure 5-19* Password Dialog Box: Generation Field and Menu

If you choose the `Choose from list` option, the Password Generator dialog box displays whenever the user chooses the `Change password` option from the Trusted Path menu. (To see the Password Generator dialog box, see Figure 5-16 on page 136).

**f. Choose from the Status Menu to Open the account, leave it Closed, or set it Always Open. (See Figure 5-20**

A new account is closed until the security administrator opens it. The status of an open account automatically changes to closed after three failed login attempts under the account's user name. An always-open account is always accessible; it is never closed as a result of multiple failed logins.

---

**Note –** When an existing account is closed after three failed login attempts, the security administrator should investigate the situation before reopening the account.

---

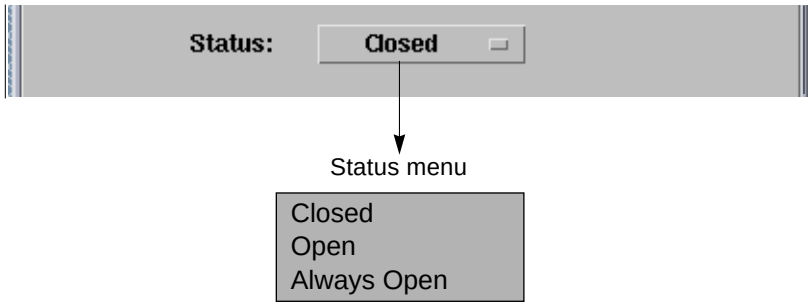



*Figure 5-20* Password Dialog Box: Status Field and Menu

**g. Click the toggle box next to Cred. Table Setup to add the NIS+ principal's public and private keys to the cred table. (See Figure 5-21.)**

*Figure 5-21* Credential Table Setup Check Box

**Note –** The recommended option is to check the Cred. Table Setup box when creating a new account so that the account is added to the NIS+ cred table and the account's password is added to the NIS+ databases, and to leave the toggle as already specified when modifying an account, unless you change the password.

**h. Store your changes by clicking Apply or OK at the bottom of the Password dialog box (see Figure 5-22).**
Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

*Figure 5-22* Controls on the User Manager: Password Dialog Box

i. **If you are done entering information for a new account or to update an existing account, go to Step 9 on page 155.**

4. **Enter a new account's home directory information, or change it for an existing account, if desired.**
   See "Home" on page 113, if necessary, to review the guidelines on what to specify in the Home dialog box.

   a. **Click Home on the User Manager Navigator.**
      The Home dialog box displays (see Figure 5-23 on page 142).

   b. **Check the Create Home Dir check box to have the account's home directory created as an MLD (optional).**

   c. **Type in the pathname for the home directory next to Path.**

   d. **Type in the name of the home directory's server next to Server.**
      If the home directory is mounted on a local partition on the account's primary workstation, enter the name of the primary workstation. Otherwise, enter the name of the NFS server for the account's home directory,

**Note –** The server must be configured prior to creation of the user account.

   e. **Check the appropriate permissions boxes under Permissions to specify the read, write, and execute permissions for the home directory by owner, group, and other.**

   f. **Specify the mail server (optional).**

   g. **Check the AutoHome Setup box to have the home directory be automounted automatically.**

*Figure 5-23* User Manager: Home Directory Dialog Box

**h. Store your changes by clicking Apply or OK at the bottom of the Home dialog box (see Figure 5-22).**
Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

*Figure 5-24* Controls on the User Manager: Home Dialog Box

**i. If you are done entering information for a new account or done updating an existing account, go to Step 9 on page 155.**

**5. Enter a new account's label information, or change it for an existing account, if desired.**
See "Labels" on page 116, if necessary, to review the guidelines on what to specify in the Labels dialog box.

**a. Click Labels on the User Manager Navigator.**
The Labels dialog box displays (see Figure 5-25).

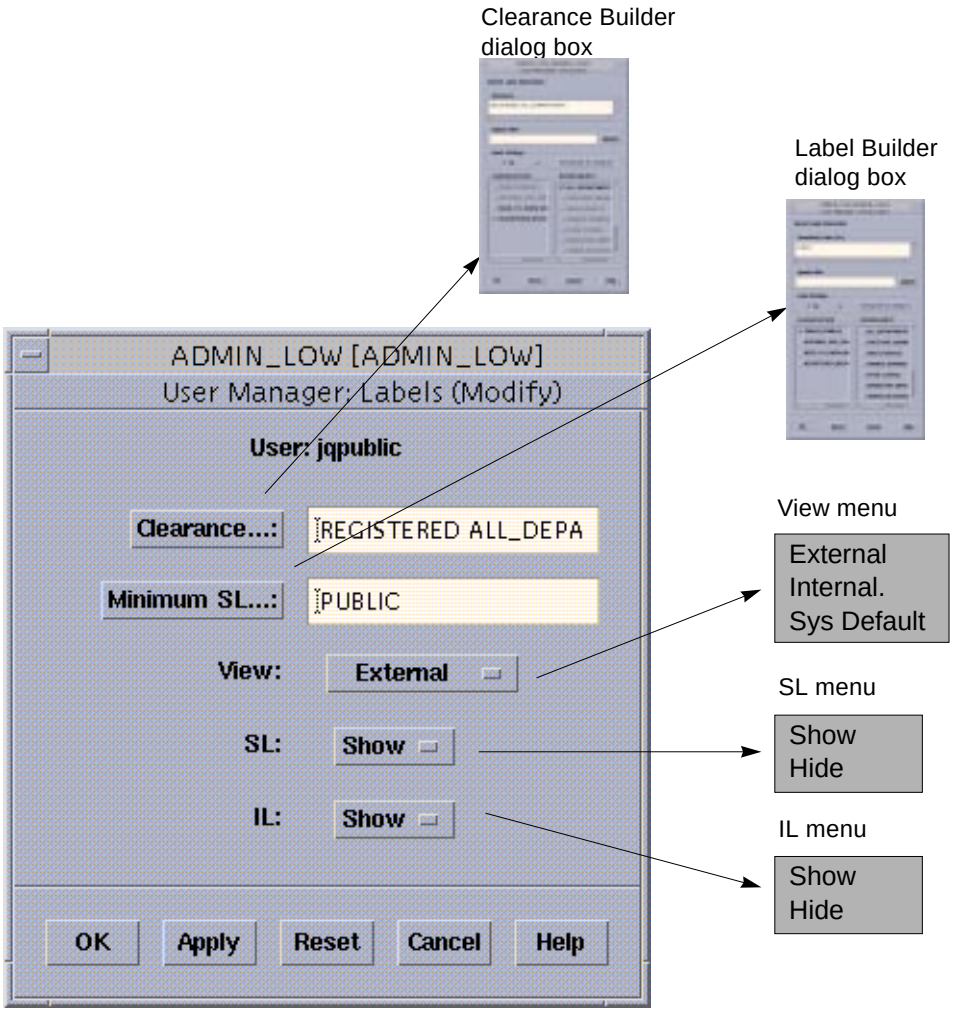



*Figure 5-25* User Manager: Labels Dialog Box

**a. Click the Clearance button to bring up the Clearance Builder dialog box and enter the account's clearance.**
The User Manager: Clearance label builder dialog box displays (see Figure 5-26 on page 145.

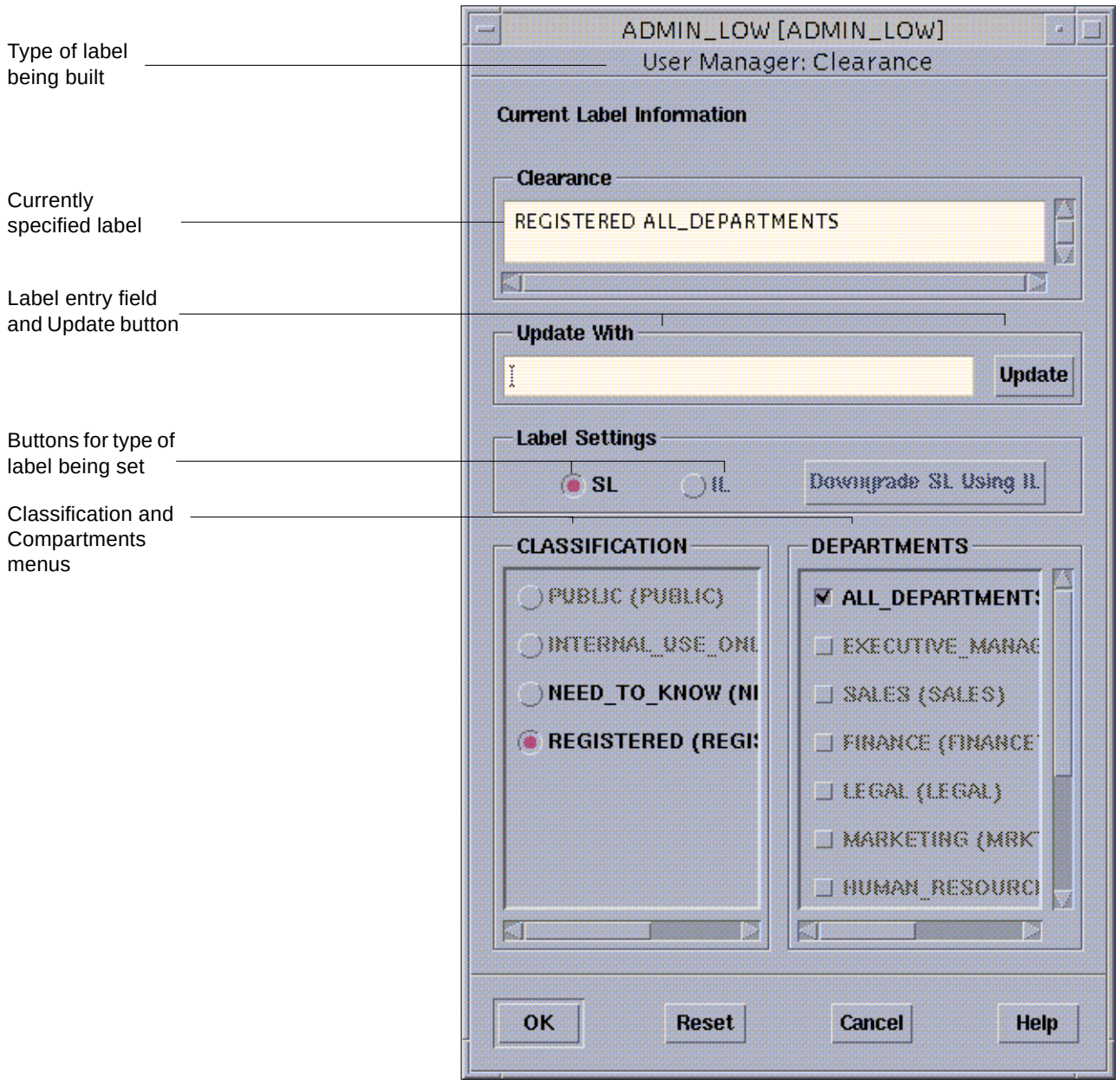



*Figure 5-26* Label Builder for Setting the Account's Clearance

i. **To type in the clearance, use the text entry field under Update With and select the Update button when you are done.**

ii. **To use the mouse to build the clearance, click a button next to the desired classification in the menu and click none or more boxes next to the desired compartment names.**

iii. **Click OK to close the label builder dialog box.**
The clearance displays in the Clearance field in the Labels dialog box.

---

**Note –** The account's clearance must be dominated by the maximum clearance and must dominate the minimum clearance specified in the `label_encodings` file.

---

b. **Click Minimum to enter the user's minimum sensitivity label.**
The Minimum Sensitivity Label Builder dialog box displays (see Figure 5-27 on page 147).

i. **To type in the minimum sensitivity label, use the text entry field under Update With and select the Update button when you are done.**

ii. **To use the mouse to build the minimum sensitivity label, check a button next to one of the classifications in the menu and check none or more boxes next to the desired compartment names by using the left mouse button.**

iii. **Click OK to close the label builder dialog box.**
The minimum sensitivity label displays in the Minimum SL field in the Labels dialog box.

---

**Note –** The minimum SL must be dominated by the maximum SL and must dominate the minimum SL defined in the `label_encodings` file.

---

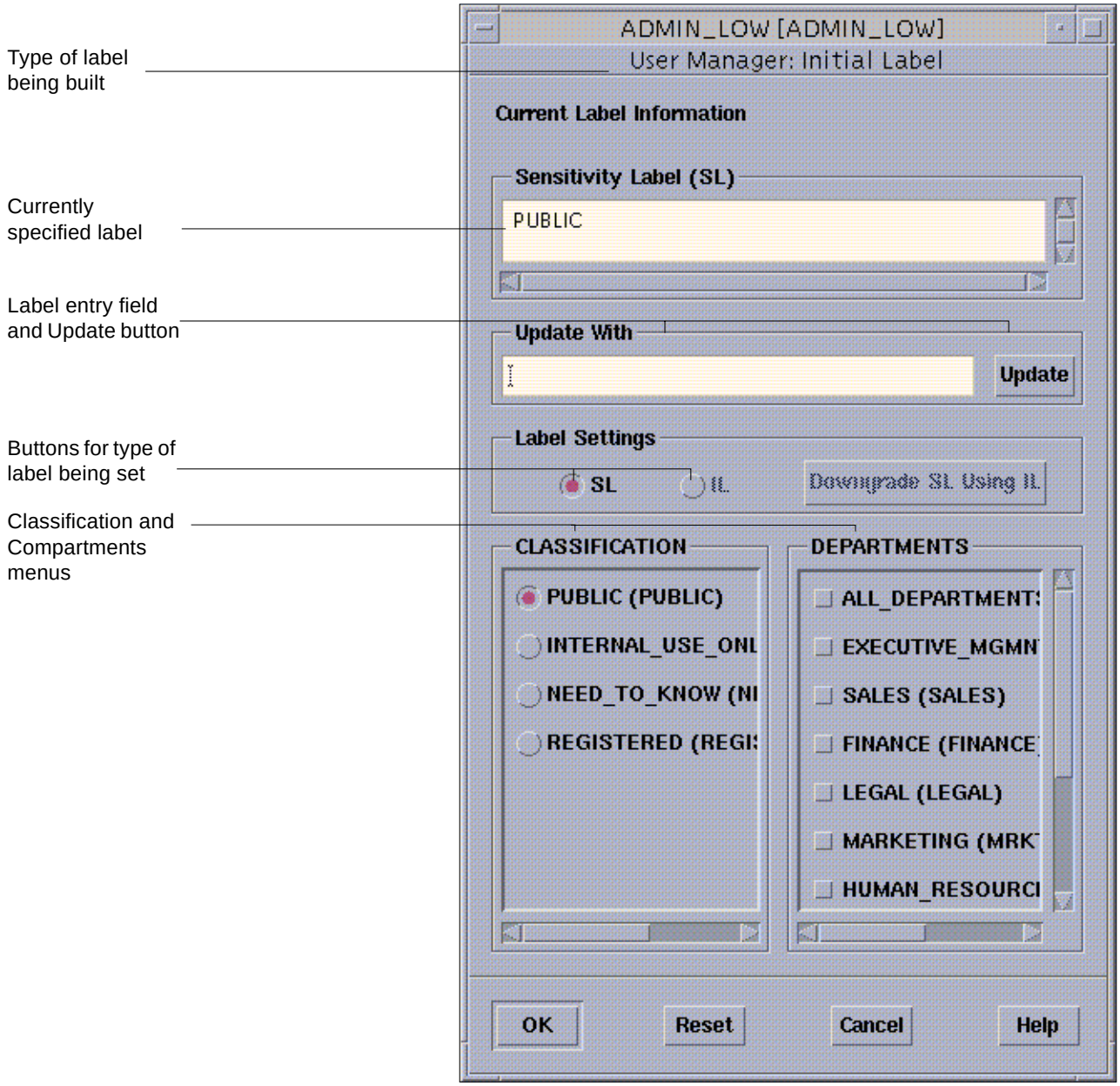



*Figure 5-27* Label Builder for Setting the Minimum SL

**c. Choose the account's view of administrative labels from the options on the View menu.**

**i. Choose External to substitute the maximum and minimum labels that are within the User Accreditation range instead of showing the ADMIN_HIGH and ADMIN_LOW administrative labels.**

**ii. Choose Internal to allow the user to see administrative labels ADMIN_HIGH and ADMIN_LOW or any site-configured alternative ASCII names.**

**iii. Choose Sys default to use the system default label view that is specified in the `label_encodings(4TSOL)` file.**

---

**Note –** The display options in the View have no effect on SLs if you do not configure the account to be able to see SLS and they have no effect on ILs if either ILs are configured not to display system-wide, or if ILs are configured to display system-wide but you do not configure the account to be able to see ILs.

---

**d. Choose whether SLs are displayed for the account from the SL menu.**

**i. Choose Show to allow SLs to be displayed.**

**ii. Choose Hide to hide SLs.**

**e. Choose whether ILs are displayed for the account from the IL menu.**

**i. Choose Show to allow ILs to be displayed.**

**ii. Choose Hide to hide ILs.**

**f. Store your changes by clicking Apply or OK at the bottom of the Labels dialog box (see Figure 5-22).**
Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

*Figure 5-28* Controls on the User Manager: Labels Dialog Box

**g. If you are done entering information for a new account or done updating an existing account, go to Step 9 on page 155.**

6. **Specify a new account's profile or profiles, or change the profiles selected for an existing account, if desired.**
   See "Roles" on page 122, if necessary, to review the guidelines on what to specify in the Profiles dialog box.

   a. **Click Profiles on the User Manager Navigator.**
      The Profiles dialog box displays (see Figure 5-29).

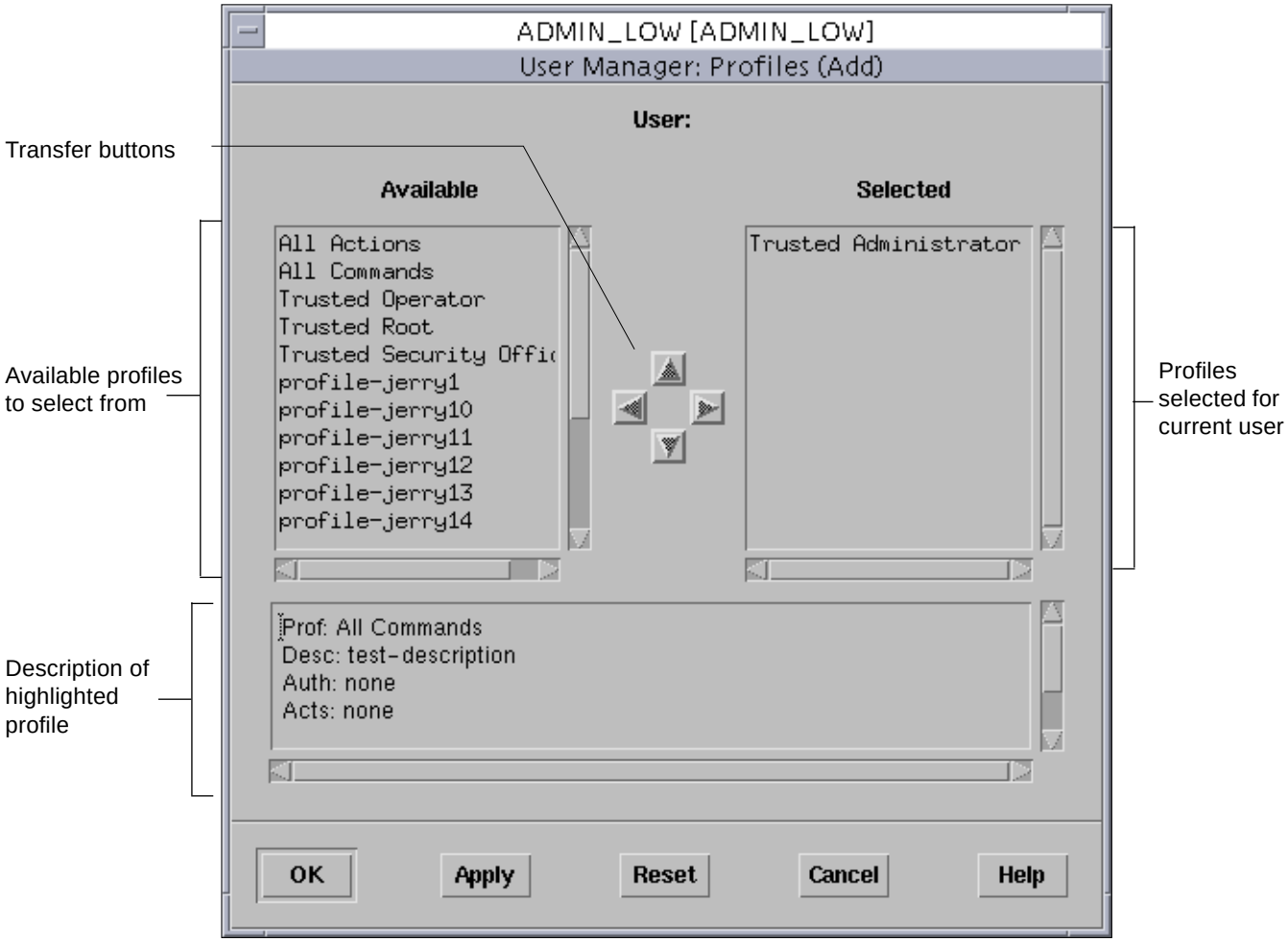



*Figure 5-29* User Manager: Profiles Dialog Box

The list at the left of the dialog box displays the available execution profiles that have not been selected. The list at the right of the dialog box contains the execution profiles that have been selected for this account.

**b. Assign an execution profile to the account either by double-clicking its name in the Available list, or by single-clicking it and then clicking the right arrow to move it into the Selected list.**

**c. Remove execution profiles from the Selected list, if desired, either by double-clicking the name of the profile or single-clicking it and then clicking the left arrow to move it to the Available list.**

**d. To change the order of an execution profile in the Selected list, single-click on the name of the execution profile and then click the up- or down arrows to move it around in the list.**

---

**Note –** As mentioned under “Profiles” on page 120, keep in mind that the order of profiles can affect which command definitions are used, and take care to have the order reflect your true intention for a user’s use of a command.

---

**e. Store your changes by clicking Apply or OK at the bottom of the Profiles dialog box (see Figure 5-30).**
Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

*Figure 5-30* Controls on the User Manager: Profiles Dialog Box

**f. If you are done entering information for a new account or done updating an existing account, go to Step 9 on page 155.**

**7. Specify any role(s) for the account in the Roles dialog box.**
See “Roles” on page 122, if necessary, to review the guidelines on what to specify in the Roles dialog box.

**a. Click Roles on the User Manager Navigator.**
The Roles dialog box displays (see Figure 5-31 on page 152).

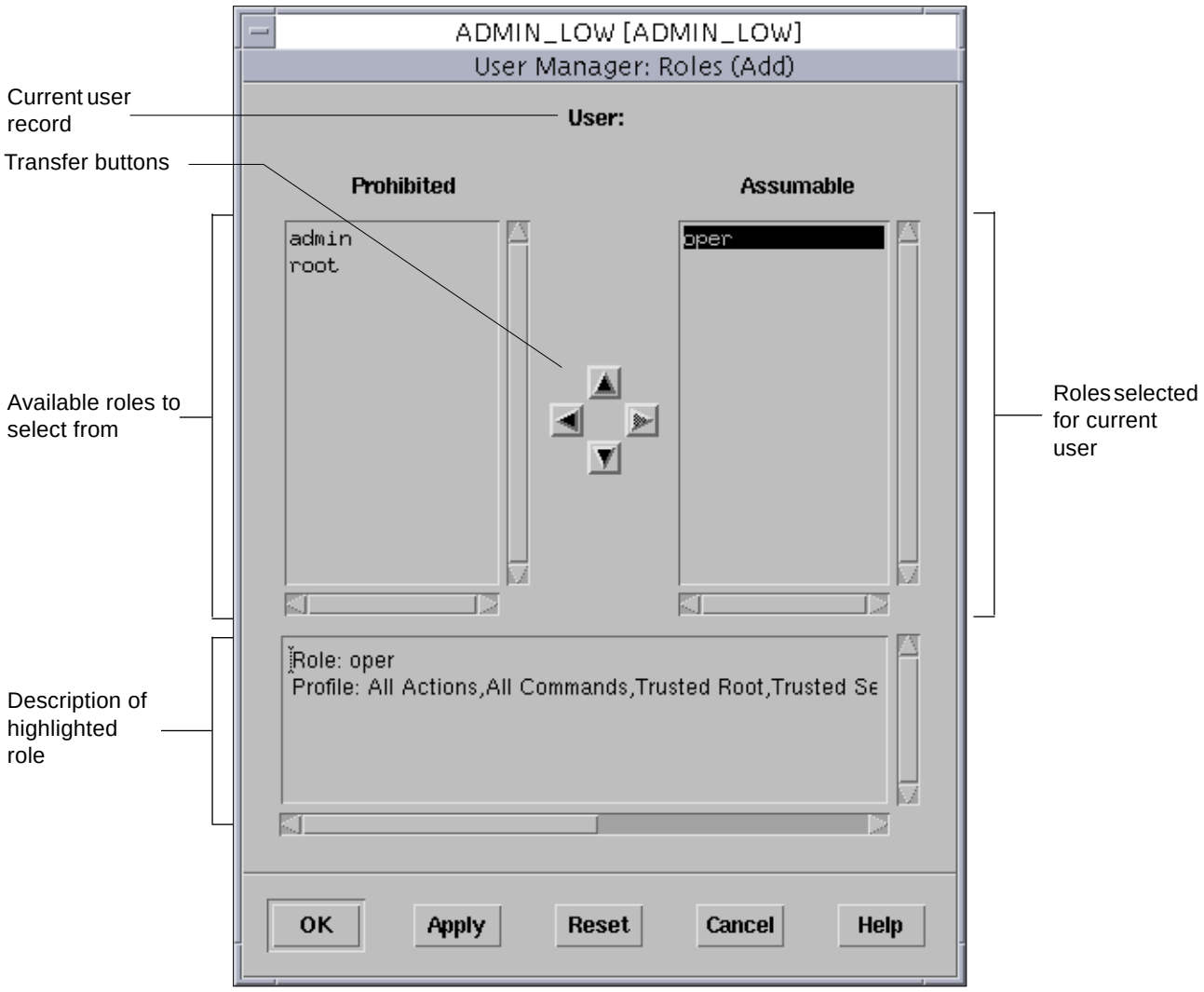



*Figure 5-31* User Manager: Roles Dialog Box

**Note –** Roles cannot assume roles, so if the Account Type field in the Identity dialog box identifies this as a role account, the Roles button is dimmed on the navigator.

The list at the left of the dialog box displays a list of roles that have not been selected. The list at the right of the dialog box under Assumable contains the roles that have been selected for this account.

**b. Assign a role to the account either by double-clicking its name in the Prohibited list, or by single-clicking it and then clicking the right arrow to move it into the Assumable list.**

**c. Remove roles from the Assumable list, if desired, either by double-clicking the name of the role or single-clicking it and then clicking the left arrow to move it to the Prohibited list.**

**d. Store your changes by clicking Apply or OK at the bottom of the Roles dialog box (see Figure 5-22).**
Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

*Figure 5-32* Controls on the User Manager: Roles Dialog Box

**e. If you are done entering information for a new account or done updating an existing account, go to Step 9 on page 155.**

**8. Specify any idle limits and actions for the account in the Idle dialog box (optional).**
See “Idle” on page 122, if necessary, to review the guidelines on what to specify in the Idle dialog box.

**a. Click Idle on the User Manager Navigator.**
The Roles dialog box displays (see Figure 5-31 on page 152).

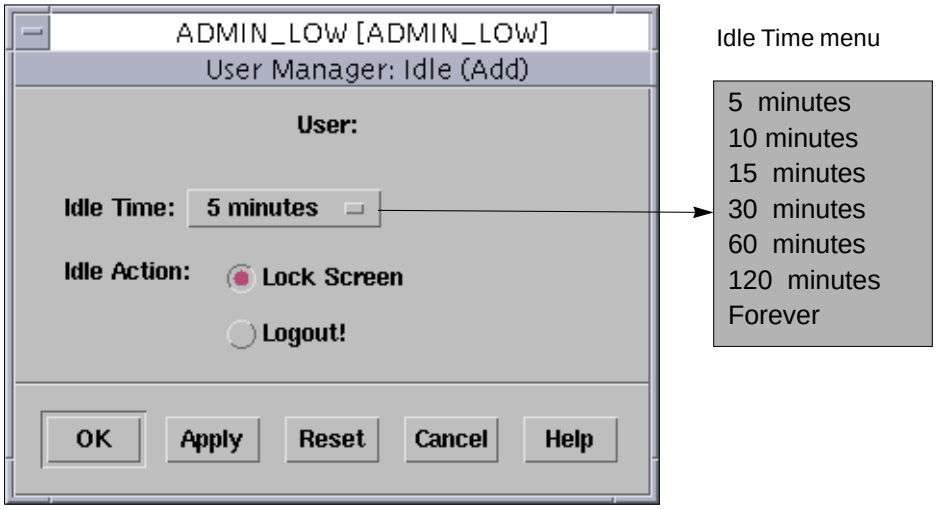



*Figure 5-33* User Manager: Idle Dialog Box with Idle Time Menu

**b. Choose from the Idle Time menu a length of time for the workstation to remain idle before an action is taken.**

**c. Click on either Lock Screen or Logout!.**

---

**Caution –** If the logout action is selected, processes running in the user's session are killed.

---

**d. Store your changes by clicking Apply or OK at the bottom of the Idle dialog box (see Figure 5-22).**
Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

*Figure 5-34* Controls on the User Manager: Idle Dialog Box

**e. If you are done entering information for a new account or done updating an existing account, go to Step 9 on page 155.**

9. **Click Done or Save on the User Manager Navigator.**
   Clicking Done in the Navigator saves what you've entered in all of the dialog boxes and closes the Navigator. Clicking Save saves what you've entered and leaves the Navigator open.

*Figure 5-35* Controls on the User Manager Navigator

**Note –** If no password information has been saved, the account stays closed.

10. **Choose the Exit option from the User Manager File menu to exit from all User Manager windows.**

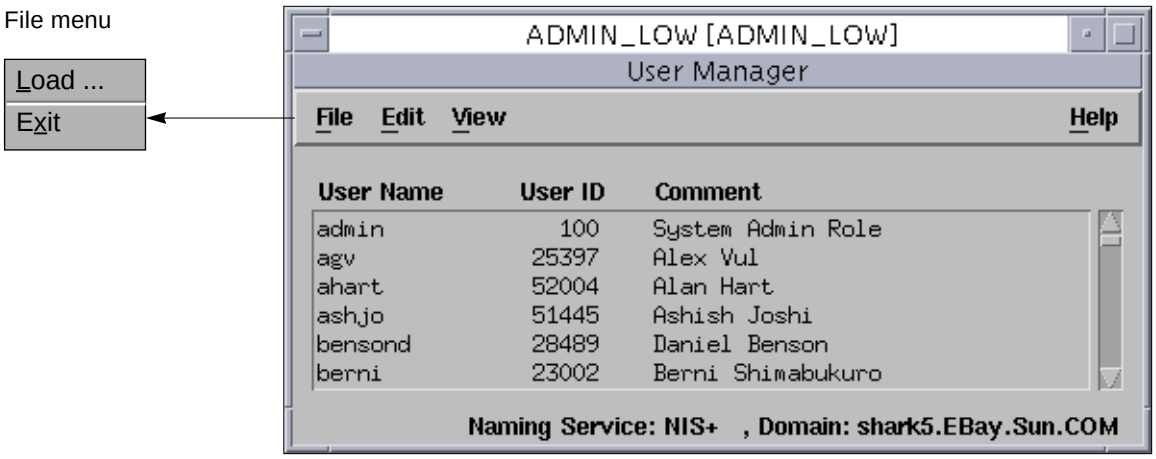



*Figure 5-36* User Manager: Main Window and File Menu

# 5

# *Managing Mail* 6

Because mail is essentially the same in the Trusted Solaris environment as it is in the Solaris environment, the system administrator role sets up and administers mail servers according to instructions in Chapter 9 of the Solaris 2.5 *User Accounts, Printers, and Mail Administration* manual. The differences in how to administer Trusted Solaris mail are described here and on the `sendmail(1MTSOL)` man page. This chapter covers the following topics:


| | |
|---|---|
| *Overview of Trusted Solaris Mail Features* | *page 158* |
| *Changing Mail Aliases* | *page 164* |
| *Enabling the Use of .mailrc Files in Home Directory MLDs* | *page 164* |
| *Creating and Initializing New Local and NIS+ Managed Aliases* | *page 166* |
| *Allowing Users to List the Entire Mail Queue* | *page 167* |
| *Tracing Sendmail's Activities* | *page 168* |
| *Troubleshooting Mail Delivery Difficulties* | *page 171* |
| *Configuring Trusted Solaris Mail Delivery Options for Mail Below Users' Minimum Labels* | *page 177* |
| *How Sendmail Handles Mail Below the Recipient's Minimum SL* | *page 177* |
| *Mail Handling Options* | *page 177* |
| *Substituting an Alternate Mail Application* | *page 180* |

# 6

This chapter contains the following procedures:


| | |
|---|---|
| *To Edit Aliases* | *page 167* |
| *To Allow Listing of the Mail Queue* | *page 168* |
| *To Trace Sendmail for Trusted Solaris Information* | *page 170* |
| *To Check for a Properly Configured Network Connection for Sending Mail* | *page 172* |
| *To Configure Mail Delivery Options for Mail Below Users' Minimum Labels* | *page 178* |
| *To Substitute an Alternate Mail Application in the Front Panel for All Users* | *page 181* |
| *To Create a Multilevel Action for the Alternate Mail Application* | *page 188* |


## *Overview of Trusted Solaris Mail Features*

In the Trusted Solaris environment, mail may be sent and received at multiple sensitivity labels by each user, and the sensitivity labels and information labels of mail messages are preserved. The following are the highlights of how mail is handled differently from the Solaris environment.

- MLDs are used to store mail messages queued for delivery and to store unread mail messages.

  See "Multilabel Directories for Outgoing and Incoming Mail" and "Mailboxes in Multilabel Directories" on page 160.

- Sendmail is modified to enforce sensitivity labels, to float information labels, and to route mail to a user only if the sensitivity label of the incoming mail is within the clearance and minimum sensitivity label of the user.

- Because users should not be able to list queued mail sent by other users, the `restrictmailq` option is set by default in the `sendmail.cf` file, and only users in the same group as the mail queue may list jobs in the mail queue.

  See "To Allow Listing of the Mail Queue" on page 168.

- The `p` option in the `sendmail` configuration file, `sendmail.cf`, has been extended with `Optsol`... privacy option settings that allow administrators to specify what `sendmail` should do with any mail that arrives at a sensitivity label below the user's minimum sensitivity label.

See “How Sendmail Handles Mail Below the Recipient’s Minimum SL” on page 177 and “To Configure Mail Delivery Options for Mail Below Users’ Minimum Labels” on page 178.

- An X-Sender-Information-Label line is added to the header of each message by `sendmail` to show the information label of the original sender.

  If an organization chooses to configure the system without information labels and, as a result, the `tsol_enable_il` switch is off in `/etc/system`, no X-Sender-Information-Label displays.

**Note –** All the fields in the message header, including the X-Sender-Information-Label are vulnerable to manipulation or spoofing by an unscrupulous user, so the information label in the X-Sender line is not strictly reliable.

- A CDE action for an alternate mail application can be substituted for the default `dtmail` action in the front panel mail subpanel.

  See “Substituting an Alternate Mail Application” on page 180.

## *Multilabel Directories for Outgoing and Incoming Mail*

The `/var/spool/mqueue` directory is an MLD where messages at varying sensitivity labels are stored until they can be delivered. Figure 6-1 shows that when a user changes to `/var/spool/mqueue`, the user is transparently redirected into `.SLD.2`, which is the SLD whose sensitivity label is the same at the current process. The mail messages shown in `.SLD.2` are the ones waiting to be sent at the current sensitivity label.

Within the mqueue MLD, a listing of all the SLDs shows:

- `.SLD.1: Permission denied` (because the label of `.SLD.1` strictly dominates the current sensitivity label))
- `.SLD.0` is empty, and

- Messages in `.SLD.2` are waiting to be sent at the current sensitivity label.

```
trustworthy% cd /var/spool/mqueue
trustworthy% mldpwd
/var/spool/.MLD.mqueue/.SLD.2
trustworthy% ls
dfNAA00212
dfNAB00212
dfNAC00212
trustworthy% cd /var/spool/.MLD.mqueue
trustworthy% ls -R .SLD.*
ls: .SLD.1: Permission denied
.SLD.0:
.SLD.2:
dfNAA00212
dfNAB00212
dfNAC00212
```

*Figure 6-1* `/var/spool/mqueue` MLD and its Contents at Different Sensitivity Label

The Trusted Solaris software also makes `/var/mail` a multilabel directory (MLD) to store mail in mailboxes for all accounts. See "Mailboxes in Multilabel Directories" for additional information.

## *Mailboxes in Multilabel Directories*

The `sendmail` program creates mailbox files within the `/var/mail` MLD to store mail received for accounts. For each account, a separate mailbox is created at each of the sensitivity labels of any mail the account has received. As a result, each account may have multiple mailboxes: if user roseanne has received mail labeled PUBLIC, NEED TO KNOW ENGINEERING and NEED TO KNOW MARKETING, she has three separate mailboxes each at a different sensitivity label in a separate SLD. The mailboxes are given the name of the account. All of user roseanne's mailboxes are called *roseanne*. Role accounts have their own mailboxes, like all user accounts. Any mailboxes created for the role account secadmin are called *secadmin*.

The first time mail is sent to any user or role on a host at a particular sensitivity label, the `sendmail` program creates an SLD in `/var/mail` at the sensitivity label of the mail and stores the mail in a mailbox in the new SLD. Subsequent mail at the same sensitivity label whether it is for another user or the same

user is put into the same SLD. When a user removes all the mail from a mailbox at a specific sensitivity label, the mailbox (`/var/.MLD.mail/.SLD.?/`*username*) is deleted.

For example, if the first mail received on the system has a sensitivity label of NEED TO KNOW ENGINEERING and is addressed to user roseanne, `/var/.MLD.mail/.SLD.0` is created at sensitivity label NEED TO KNOW ENGINEERING, and the mail is put into the `roseanne` mailbox in `.SLD.0`. If the next time mail is received on the system, it is also has the NEED TO KNOW ENGINEERING sensitivity label and is addressed to user jhoman, this second piece of mail is put into `.SLD.0` in the `jhoman` mailbox. A third piece of mail to roseanne at PUBLIC causes `.SLD.1` to be created; the mail is put into the `roseanne` mailbox in `.SLD.1`. Figure 6-2 shows that the mail sent to roseanne and jhoman at NEED TO KNOW ENGINEERING is being stored in the `roseanne` and `jhoman` mailboxes in `.SLD.0`, while mail sent to user roseanne at PUBLIC is in another `roseanne` mailbox in `.SLD.1`.

```
$ cd /var/.MLD.mail/
$ ls -R .SLD.*
.SLD.0
roseanne
jhoman
.SLD.1
roseanne
```

*Figure 6-2* Mailboxes in SLDs at Different Sensitivity Labels

## Mail Notification

Accounts are notified about mail arrival through the Mail subpanel on the Front Panel in whatever workspace is active. As illustrated in Figure 6-3, the Mail subpanel displays a mail icon for each of the following:

- For each sensitivity label at which mail has been received by the logged-in user account
- For each sensitivity label at which mail has been received by any role that the user has assumed during the current session.

*Figure 6-3* Mail Subpanel With Mail at Multiple Labels

Figure 6-3 shows the Mail subpanel displayed in a workspace with the INTERNAL_USE_ONLY sensitivity label. The Mail subpanel shows mail icons at PUBLIC, INTERNAL_USE_ONLY, and NEED_TO_KNOW for the user. The

The Mail subpanel in Figure 6-3 on page 162 also shows mail at ADMIN_LOW for the root role, which the user has assumed during the session—as indicated by the presence of a root administrative role workspace button at the lower right of the figure.

The mail icon at the top of a subpanel is not identified with a sensitivity label. If clicked, it displays with the sensitivity label of the current workspace, just like the mail icon in the Front Panel.

Access to mail at various sensitivity labels is not constrained by the sensitivity label of the current workspace, because it does not violate the Trusted Solaris security policy for any user to be able to read and send mail at any sensitivity label for which the user is cleared. Therefore *a mail reader may be launched at any of the sensitivity labels of any of the icons displayed in the mail subpanel, no matter what the sensitivity label is of the current workspace.*

## *Reading of Mail*

When the recipient of the mail clicks on a mail icon, a mail reader is displayed at the sensitivity label of the incoming mail. When a complete CMW label is displayed, the short name of the sensitivity label is displayed in brackets. Because the sensitivity label of the mail shown in Figure 6-4 is INTERNAL_USE_ONLY the mail reader shows a sensitivity label of [INTERNAL] to the right of the PUBLIC information label.

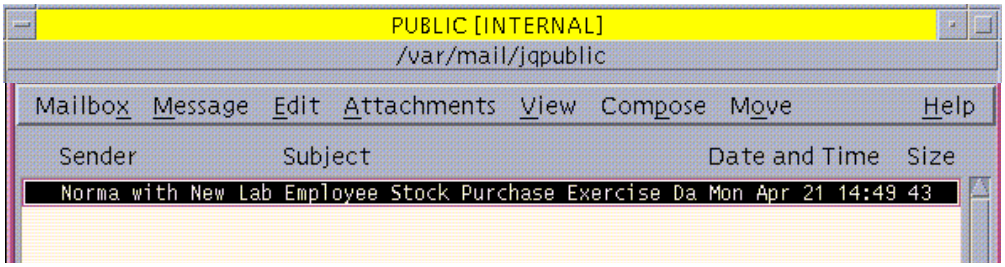



*Figure 6-4* Window Label on a Mail Reader Launched at a Sensitivity Label of INTERNAL_USE_ONLY When Information Labels are Enabled

When information labels are disabled, the long name of the sensitivity label displays within brackets. The mail reader shown in Figure 6-5 displays with a sensitivity label of [INTERNAL_USE_ONLY].

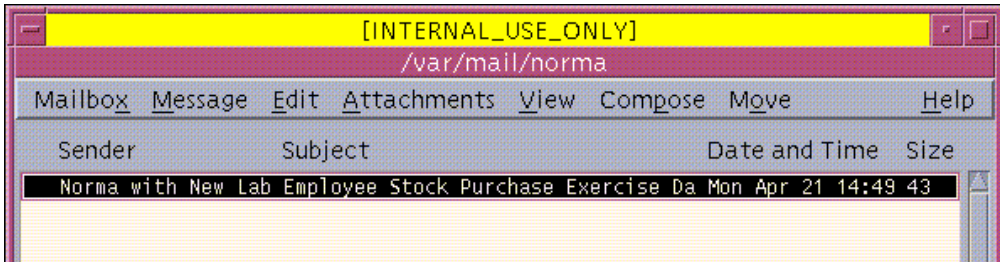



*Figure 6-5* Window Label on a Mail Reader Launched at a Sensitivity Label of INTERNAL_USE_ONLY When Information Labels are Disabled

### *How Mail Gets Its Sensitivity Label*

The sensitivity label of the mail is determined by one of the following:

- For mail coming from a Trusted Solaris or from another type of labeled host configured with one of the labeled host options in the Trusted Networking databases, the sensitivity label is the sensitivity label of the creating process (the sensitivity label of the mailer that sent the mail).
- For mail coming from an unlabeled machine (one that does not recognize labels and is configured as an unlabeled machine in the Trusted Networking databases), it is the default sensitivity label assigned to the host in the Trusted Networking databases.

## *Changing Mail Aliases*

See the Solaris 2.5 “Mail Administration Guide” for how to administer mail aliases, along with the following sections that explain the exceptions that apply in the Trusted Solaris environment.

### *Enabling the Use of* `.mailrc` *Files in Home Directory MLDs*

As in Solaris systems, individual users may put local copies of a `.mailrc` file into their home directories to create personal mail aliases. However, because Trusted Solaris home directories are almost always MLDs, the system administrator or individual user has to do some setup so that `.mailrc` files (and possibly other startup files) can be copied or linked into every SLD created within the user’s home directory at each sensitivity label. See also “Managing Startup Files in a Trusted Solaris System.”

To partially automate the process of propagating .mailrc files (and other startup files) the system administrator can make use of standard skeleton directories and two new Trusted Solaris files explained below:

- .copy_files and
- .link_files

## .copy_files

Create a .copy_files file and list the .mailrc file in it if you want the .mailrc to be copied to each SLD. (Copied startup files can be different in each SLD.) Copying the .mailrc, for example, would allow users to potentially have differing mail aliases in each SLD.

## .link_files

Create a .link_files file and list the .mailrc file in it if you want the .mailrc to be linked to each SLD. Linked startup files are the same in each SLD. Linking the .mailrc, for example, would mean that any change made to the .mailrc in any SLD would be made in all SLDs.

## *Using the* .copy_files *and* .link_files *Along With Skeleton Directories*

If the system administrator specifies a skeleton path directory for a user account, all the files in that skeleton directory are copied into the first home directory SLD created for the user. The system administrator can take advantage of this feature by putting into a skeleton directory (such as /etc/skel) a basic .mailrc along with a .copy_files that lists the .mailrc for copying or a .link_files file that lists the .mailrc for linking. Once this has been set up, and once the .copy_files or .link_files has been copied into the home directory SLD created at the user's minimum sensitivity label, if the .mailrc file is listed in .copy_files, it is copied, and if the .mailrc files is listed in .link_files, it is linked into each subsequently-created SLDs whenever the account labels a new workspace with a new sensitivity label. The user may also create a .mailrc, .copy_files, or a .link_files or edit any existing files that have been propagated to any of their home directory SLDs after the system administrator's intervention.

**Note –** Do any setup for the `.mailrc` using `.copy_files` or `.link_files` you want to make immediately *before users have a chance to work at multiple sensitivity labels*. Otherwise, multiple home SLDs may be created without startup files being propagated.

"Controlling Which Startup Files Are Read While the Window System is Coming Up" on page 65 in Chapter 3.

### ▼ To Propagate a `.mailrc` to All Accounts' Home Directory SLDs

1. **Assume the system administrator role.**
2. **In an ADMIN_LOW workspace, go to `/etc/skel` or whatever skeleton directory location you wish to use.**
3. **Put a basic `.mailrc` file into the skeleton directory.**
4. **To make the `.mailrc` copied to all home directory SLDs, create a `.copy_files` or modify it, if it already exists, to list the `.mailrc`.**
5. **To make the `.mailrc` linked to all home directory SLDs, create a `.link_files` or modify it, if it already exists, to list the `.mailrc`.**
6. **When setting up user accounts, if you want to support the use of personal mail aliases enter the pathname of the skeleton directory in the Home dialog box on the User Manager.**

## *Creating and Initializing New Local and NIS+ Managed Aliases*

Aliases may be specified in NIS+ maps or in the `/etc/aliases` file on each host. The Naming Service menu on the Trusted Solaris version of the Solstice Database manager allows the system administrator chose between NIS+ or none when editing any database, including Aliases. In the Solaris 2.5 operating system, the `aliases`(4) database is automatically initialized after any changes whenever `sendmail` delivers mail and discovers that the `aliases` database is out of date with respect to `/etc/mail.aliases`. In Trusted Solaris systems, automatic initialization of the `aliases`(4) database is only possible when `sendmail` is running at ADMIN_LOW because the `aliases` database is stored at ADMIN_LOW. When `sendmail` is running at any other sensitivity label the file cannot be written to by `sendmail`.

**Note –** The system administrator role must remember to run `newaliases` or `sendmail` with the `-bi` option after making any changes to the aliases.

### ▼ To Edit Aliases

1. **Assume the system administrator role, go to the Solstice_Apps folder in the Application Manager.**
   a. **Click the Application Manager in the front panel.**
   b. **Double click the Solstice_Apps folder in the Application Manager.**
   c. **Double click the Database_Manager icon.**
2. **Bring up the Database Manager: Aliases Database.**
   a. **Highlight Aliases on the Databases scrolling list.**
   b. **From the Naming Service menu, pick NIS+ to edit the NIS+ aliases map or pick None to edit the local version /etc/aliases file.**
   c. **Click OK.**
   The Database Manager: Aliases Database displays.
3. **To modify an alias, select the name of an existing alias and modify it.**
   a. **Highlight an alias name.**
   b. **Choose Modify from the Edit menu.**
   The Database Manager: Modify dialog displays.
   c. **Change the text in the Expansion text entry field as desired.**
4. **To add an alias, select Add from the Edit menu.**
   a. **Enter the alias name in the Alias text entry field.**
   b. **Enter all addresses for the alias in the Expansion text entry fields.**

## *Allowing Users to List the Entire Mail Queue*

In the default configuration, users are not allowed to list queued mail sent by other users. In the Solaris versions of in the `/etc/mail/sendmail.cf` file, the `restrictmailq` option is available to be used but it is not set by default.

In the Trusted Solaris implementation, `restrictmailq` is set by default in the `sendmail.cf` file, and only users in the same group as the mail queue may list jobs in the mail queue.

---

**Note –** Even if an administrator performs one of the steps described below to allow listing of the mail queue, only users who have the `mailq` or `sendmail` commands in one of their profiles can list the mail queue—by entering either **`mailq`** or **`sendmail`** with the **`-bp`** option. Also, these commands show mail only at labels dominated by the user's process.

---

### ▼ To Allow Listing of the Mail Queue

1. **To allow all users to list the mail queue, remove the `restrictmailq` option in the `/etc/mail/sendmail.cf` file.**
2. **To allow a group of users to list the mail queue:**
   a. **Create a new group.**
   b. **Add the desired set of users to the group.**
   c. **Use `chgrp(1)` to change the group of `/var/spool/mqueue` to the new group.**

## *Tracing Sendmail's Activities*

Multiple instances of `sendmail` are involved in local and remote mail delivery. Figure 6-6 on page 169 shows how data flows through the `sendmail` processes. Any mailer used to send mail (the default is `dtmail`) starts an instance of `sendmail`. This instance of `sendmail` attempts to deliver mail that originates on the host, storing it in the local `/var/spool/mqueue` MLD until it is delivered—in case the system crashes or anything else causes the mail message not to be delivered [① in Figure 6-6]. 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 a long time.

An instance of the `sendmail` program starts when the workstation or server is booted, and 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 Figure 6-6]. A third instance of `sendmail` periodically scans the mail queue and attempts to

deliver any mail in the queue [② and ④ in Figure 6-6]. Figure 6-6 shows some of the sendmail processes on three hosts: cascade, trustworthy, and juggle. Host trustworthy is the mail relay host for juggle.

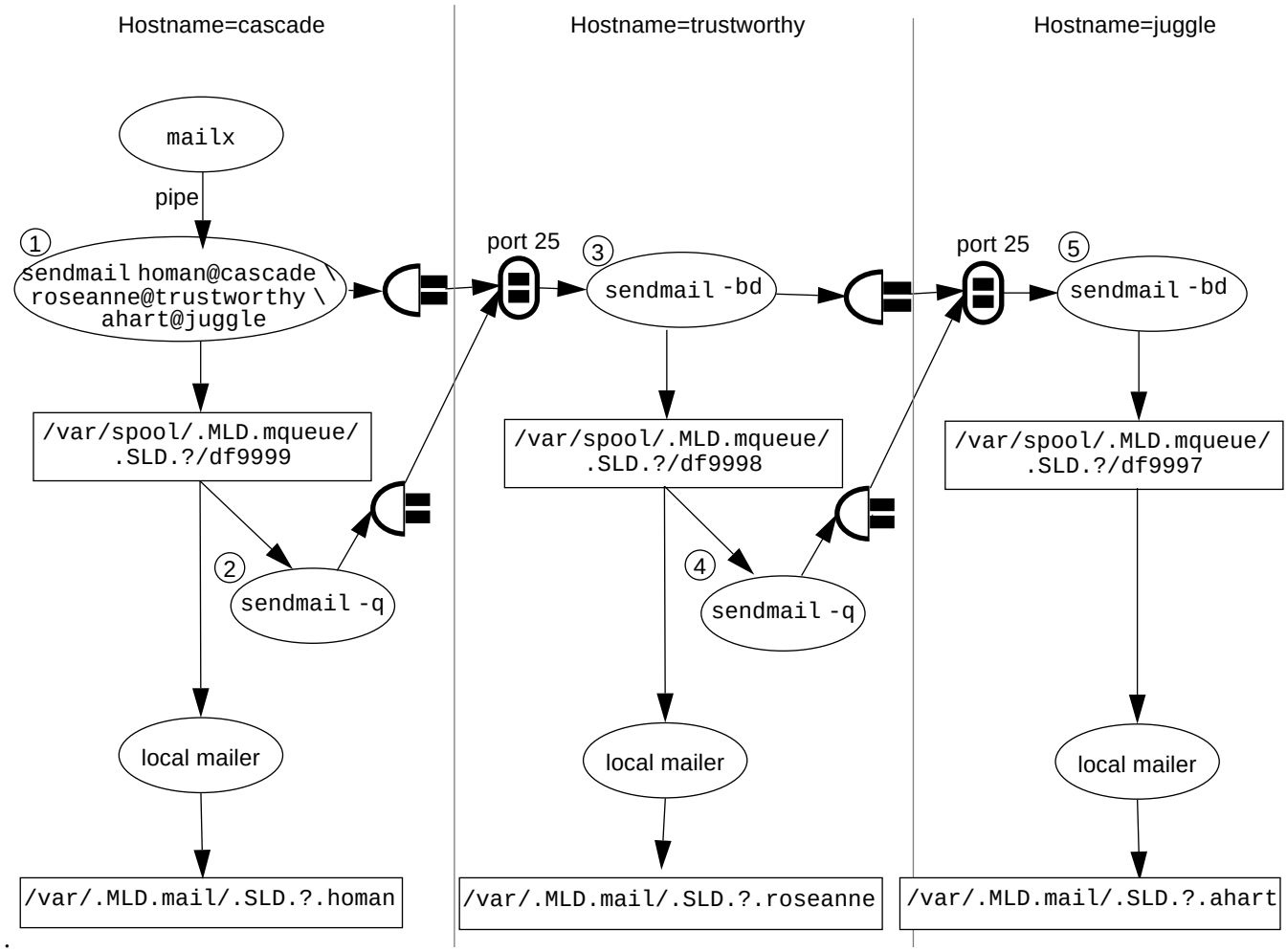



*Figure 6-6* Sendmail Data Flow Example

When you send mail to *username@hostname* and *hostname* is the name of a remote host, `sendmail` forwards the message to port 25 of *hostname*. As shown in Figure 6-6 on page 169, 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 using sendmail -d is described in detail in the sendmail Nutshell Handbook put out 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. A new category, 75, selects Trusted Solaris debugging information.

## ▼ To Trace Sendmail for Trusted Solaris Information

**1. Assume an administrative role.**

**Note –** To use sendmail with the -d option requires the trusted path attribute, which means that only administrative roles that have this command in the role's execution profile can debug sendmail.

**2. To debug mail on a remote machine, kill the current sendmail daemon.**

```
$ rlogin trusted
$ /usr/bin/ps -sv | grep sendmail
 root   210    1  0   Aug 12 ?      0:02 /usr/lib/sendmail -bd -q1h
$ /usr/bin/kill -9 210
```

3. **Enter `sendmail` with the `-d` option followed by the category `75` followed optionally by a dot (.) and a level, followed by a space and the address, followed by a message.**
   The message can either be included 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 `mail(1)`.

```
$ /usr/lib/sendmail -d75.9 roseanne@trusted < /etc/motd
```

4. **Review the error messages.**

## *Troubleshooting Mail Delivery Difficulties*

Remember that `sendmail(1MTSOL)` makes a number of checks about the destined recipient and about the destined receiving host before sending or forwarding mail. Mail can be received by an account only if the mail is within that account's sensitivity label range (between the account's clearance and minimum sensitivity label). The account's sensitivity label range is specified as described in Chapter 5, "Using the User Manager to Configure User and Role Accounts." Mail can be received on a host only if the mail is within the accreditation range of that host, as described in the following list and in Chapter 10, "Specifying Security Attributes in Trusted Network Databases and Setting Up Routing."

- Multilevel hosts with an accreditation range between ADMIN_LOW and ADMIN_HIGH can receive mail at all sensitivity labels,
- Multilevel hosts that have a restricted accreditation range can receive mail only between their maximum and minimum sensitivity labels
- Single-level (unlabeled) hosts can receive mail at only the single sensitivity label set up for them.

If a user is having trouble sending mail, use the following as guidelines for where to look:

- **Check the mail aliases.**
  Remember that the `/etc/aliases` file and the `mail_aliases` NIS+ map are consulted by `sendmail` in determining where to deliver mail. For example, mail to `fred` from a system process on his Trusted Solaris workstation `xxx` would not go to `fred@xxx` if sendmail consults the `mail_aliases` table and finds an alias of `fred@yyy` for user fred.
- **Make sure that there is a properly configured network connection between the sending and receiving hosts.**
  Do the procedure described in "To Check for a Properly Configured Network Connection for Sending Mail" on page 172.

## ▼ To Check for a Properly Configured Network Connection for Sending Mail

1. **Send mail using `mailx`.**

```
# mailx -v somebody@somehost
Subject: test1
testl
.
```

   Look at the messages from `mailx`. If you see a line that says, message accepted, go to Step 3.

2. **While logged into the sending host or, if the mail server is not the same as the sending host, while logged into the mail server, at the sensitivity label at which the user needs to send mail, use the `telnet(1)` command to connect to port 25 of the receiving host.**

```
trustworthy% telnet hostname 25
```

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

```
220 hostname Sendmail version ready at date
```

Enter quit to end the connection.

```
quit
```

If you get an error message from `telnet`, the connection is not properly set up with the correct labels in the trusted networking databases; go to Step 5 and following and do the step that applies to the type of host you are trying to debug. If the connection seems to be set up properly, go to Step 3.

3. **At the sensitivity 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.**
   Check the list to see if the mail is stuck on the mail server.

```
# mailq | more
```

4. **Try the procedure under "To Trace Sendmail for Trusted Solaris Information" on page 170.**

5. **If the destination host is a Trusted Solaris 2.x system, do these steps to make sure the destined user is able to receive mail within Trusted Solaris security policy:**

   a. **Make sure the destined recipient has a valid user account by using the User Manager.**

   b. **Note the account's minimum label and clearance in the Labels dialog in the User Manager.**

   c. **Make sure that the sensitivity label of the mail being sent is dominated by the recipient's maximum label and dominates the recipient's minimum label.**

      i. **If the sensitivity label of the mail being sent is not in the recipient's account label range, if you can find a mutually-acceptable sensitivity label for the sender and the recipient, change the sensitivity label to one within the destined recipient's label range and try again.**

      ii. **If the mail goes through, instruct the sender to send mail to that recipient at the mutually-acceptable label.**

**d. Make sure that the sensitivity label of the mail is within both the User Accreditation Range and the System Accreditation range of the destination host as defined by the `label_encodings` file.**
`sendmail` does not deliver mail if the sensitivity label of the mail is outside the System Accreditation Range.

If the sensitivity 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, remember that a normal user by default cannot receive mail sent outside of the User Accreditation Range, and go on to Step e.

**e. If the mail is below the minimum label of the destined user, do any of the following steps, if it is consistent with your site's security policy.**
See "How Sendmail Handles Mail Below the Recipient's Minimum SL" on page 177 and "To Configure Mail Delivery Options for Mail Below Users' Minimum Labels" on page 178.

**i. To enable all users on the system to receive mail that is below their minimum labels, either make sure that `sendmail` is automatically upgrading the mail by specifying `tsoluserlowupgrade` (the default), or make sure the mail reader delivers the mail at the incoming label, which allows the upgrade if the account has the needed authorization, by specifying `tsoluserlowaccept`.**

**ii. To allow the normal user accounts of all employees who are able to assume administrative roles to receive mail from system processes outside the User Accreditation Range, make sure that the `tsoladminlowupgrade` or `tsoladminlowaccept` options are set in the `sendmail` configuration file.**

**iii. To enable one administrative role to receive mail from system processes outside the User Accreditation Range, use the Profile Manager to make sure that the administrative role has the *use all defined labels* authorization.**
(The default administrative roles have that authorization in their profiles.)

**6. For a destination host running the Trusted Solaris 2.5 software, on the sending host make sure that the `tnrhdb/tnrhtp` entries for the receiving host are configured properly so the TS2.x system can communicate with that host over the network.**

**Note –** You can use the `tninfo(1MTSOL)` command to check which template has been assigned which host and which host type and other attributes are assigned in the template. 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(4TSOL)` database and that the template in the `tnrhtp(4TSOL)` file correctly defines the host's type as `sun_tsol`.**

   b. **Check that the minimum and maximum sensitivity label set in the assigned template in the `tnrhtp(4TSOL)` allow communications at the sensitivity label of the mail that is not being delivered.**

   c. **Once these checks are passed, the network connection ought to work. Go back and do Step 2 on page 172 and run `telnet` to make sure.**

7. **For a destination host running the Trusted Solaris 1.x software, on the sending host make sure that the `tnrhdb/tnrhtp` entries for the receiving host are configured properly so the TS2.x system can communicate with that host over the network.**

   a. **Check that the destination host has the correct template name assigned to it in the `tnrhdb(4TSOL)` database and that the template in the `tnrhtp(4TSOL)` file correctly defines the host's type as `maxsix`.**

   b. **Check that the minimum and maximum sensitivity label set in the assigned template in the `tnrhtp(4TSOL)` allow communications at the sensitivity label of the mail that is not being delivered.**

   c. **Once these checks are passed, the network connection ought to work. Go back and do Step 2 on page 172 and run `telnet` to make sure.**

8. **For a destination host running any labeled operating system that is not the Trusted Solaris operating system, on the sending host make sure that the `tnrhdb/tnrhtp` entries for the receiving host are configured properly so the TS2.x system can communicate with that host over the network.**

a. **Read the `tnrhtp(4TSOL)` 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 RIPSO type hosts require other options. See also Chapter 10, "Specifying Security Attributes in Trusted Network Databases and Setting Up Routing."

b. **Create a template or copy an applicable one in the `tnrhtp(4TSOL)` and make sure that the correct template is assigned to the host in the `tnrhdb(4TSOL)` database to identify it with the appropriate host type.**

c. **Check that the minimum and maximum sensitivity label set in the assigned template in the `tnrhtp(4TSOL)` allow communications at the sensitivity label of the mail that is not being delivered.**

d. **Once these checks are passed, the network connection ought to work. Go back and do Step 2 on page 172 and run `telnet` to be sure.**

9. **If the destination host is not running a label-cognizant operating system, on the sending host make sure that the `tnrhdb/tnrhtp` entries for the receiving host are configured properly so the TS2.x system can communicate with that host over the network.**

   a. **Check that the destination host has the correct template name assigned to it in the `tnrhdb(4TSOL)` database that the template in the `tnrhtp(4TSOL)` file correctly defines the host's type as `unlabeled`.**

   b. **Check that the sensitivity label defined as the default sensitivity label for the unlabeled host in the assigned template in the `tnrhtp(4TSOL)` allows communications at the sensitivity label of the mail that is not being delivered.**

   c. **Once these checks are passed, the network connection ought to work. Go back and do Step 2 on page 172 and run `telnet` to be sure.**

## *Configuring Trusted Solaris Mail Delivery Options for Mail Below Users' Minimum Labels*

The security administrator must make sure that the Trusted Solaris-specific privacy options in the `sendmail` configuration file `sendmail.cf` have values consistent with the site's security policy. See the `sendmail(1MTSOL)` man page, especially the TRUSTED SOLARIS DIFFERENCES section, and see the paragraphs that follow for a description of what the options do.

## How Sendmail Handles Mail Below the Recipient's Minimum SL

Two types of actions may be specified for `sendmail` to take when mail is received at a sensitivity label that is below the recipient's minimum sensitivity label. The `Optsol`... privacy option settings in the `sendmail` configuration file specify the actions to take, depending on whether the mail is at ADMIN_LOW or at another label below the recipient's sensitivity label. 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, while a user cleared to a particular label in the user accreditation range, such as CONFIDENTIAL or INTERNAL USE ONLY), should probably not be able to send mail to a user whose minimum label is SECRET or NEED TO KNOW.

These options give sites the discretion to decide which response is consistent with their site's security policy. The defaults (as set in `/etc/mail/sendmail.cf`)automatically upgrade mail sent at ADMIN_LOW and return mail sent at other labels below the user's minimum sensitivity label.

### Mail Handling Options

Three mutually exclusive options have the prefix *tsoladminlow* and the three other mutually exclusive options have the prefix *tsolother*. Each option has an action portion in its name to specify what `sendmail` should do when it receives the mail at ADMIN_LOW or some other sensitivity label below the user's minimum sensitivity label. *upgrade* means to deliver the message at the

recipient's minimum sensitivity label. *accept* means to deliver the message at the message's sensitivity label. *return* means to return the message to the sender.

*Table 6-1* Trusted Solaris Mail Handling Options

| Option Name | Effect |
|---|---|
| `tsoladminlowupgrade` | Default setting. Accepts ADMIN_LOW mail and delivers it at ADMIN_LOW. User may upgrade it and read it only if one of the user's execution profiles has the SYS_ACCRED_SET authorization. |
| `tsoladminlowaccept` | Accepts ADMIN_LOW mail and delivers it at ADMIN_LOW. User may upgrade it and read it only if one of the user's execution profiles has the SYS_ACCRED_SET authorization. |
| `tsoladminlowreturn` | Returns ADMIN_LOW mail to the sender |
| `tsolotherlowupgrade` | Upgrades mail received at a SL below the user's minimum SL to the user's minimum SL. Because the mail is upgraded by `sendmail`, the user does not need the SYS_ACCRED_SET authorization to receive it. |
| `tsolotherlowaccept` | Accepts mail below the user's minimum label and delivers it. User may upgrade and read it only if one of the user's execution profiles has the SYS_ACCRED_SET authorization. |
| `tsolotherrlowreturn` | Default setting. Returns mail below the user's minimum label to the sender. |

## ▼ To Configure Mail Delivery Options for Mail Below Users' Minimum Labels

1. **As security administrator, use the Set Mail Options action to open the `sendmail.cf` file for editing.**
   a. **Click the application manager icon on the front panel.**
   b. **In the Application Manager folder, double-click the System_Admin icon.**
   c. **Double-click the Set Mail Options action icon.**

2. **Search for the lines that begin `Optsol`, and change either of the two existing default settings.**

See Table 6-1 on page 178, ““Trusted Solaris Mail Handling Options” for names and descriptions of the tsol privacy options.

```
# TSOL actions for mail received below recipient min label
#    options are:
#       tsoladminlowupgrade - upgrade to user min label (default)
#        tsoladminlowaccept  - accept at delivered label
#        tsoladminlowreturn  - return to sender
#        tsolotherlowupgrade - upgrade to user min label
#        tsolotherlowaccept  - accept at delivered label
#        tsolotherlowreturn  - return to sender (default)

Optsoladminlowupgrade
Optsolotherlowreturn
```

# 6

## *Substituting an Alternate Mail Application*

By default, `dtmail` is the mail application that is launched from the mail panel the Trusted Solaris Front Panel. The Trusted Solaris system allows the substitution of an alternate mail application, but only the system administrator can do the set up needed so that the mailer provides the 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 sensitivity label of the current workspace. However, since mail monitoring for mail at multiple sensitivity 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 sensitivity label.

The system administrator can either

- Modify the front panel control file so that an alternate mail action is available to all users (see “To Substitute an Alternate Mail Application in the Front Panel for All Users”).
- Make an alternate mail action control file available with instructions for individual users on how to drag and drop the alternate control file into their Front Panel mail subpanel (see “To Create a Multilevel Action for the Alternate Mail Application”).

For an alternate mail action to be installed in the front panel, the mail application must have an action defined. The example screens in “To Substitute an Alternate Mail Application in the Front Panel for All Users” on page 181

show the substitution of the OpenWindows `mailtool` for `Dtmail`. The OpenWindows `mailtool` action is defined in the `/usr/dt/appconfig/types/C/sunOW.dt` file as shown in Figure 6-7.

```
ACTION OWmailtool
{
        LABEL           OW Mail Tool
        ICON            OWmailtool
        TYPE            COMMAND
        WINDOW_TYPE     NO_STDIO
        EXEC_STRING     /usr/openwin/bin/mailtool
}
```

*Figure 6-7* OpenWindow's `mailtool` Action Definition from `sunOW.dt`

*Tip*

**Note –** Difficulties may arise if mail arrives while you are installing an alternate mail icon or deleting the default one. For that reason, it is a good idea to stop sendmail before you start and start sendmail again after you are done.

If all the mail icons disappear from the Front Panel, investigate the account's $HOME`/.dt/fp.dynamics` directory. During the operation of the system, all changes to the Front Panel are stored in each account's $HOME`/.dt/fp.dynamics` directory at the session clearance. Copy the contents of `fp.dynamics` to a backup directory and restore the file one by one until the Front Panel configuration is restored.

### ▼ To Substitute an Alternate Mail Application in the Front Panel for All Users

**Caution –** Do this procedure before accounts start getting mail on the system. If you do it later, you would need to clean up the contents of directories created by the window system in every $HOME`/.dt/fp.dynamics` directory.

1. **Assume the security administrator role and make sure you are in an ADMIN_LOW workspace.**

2. **Make sure an action is defined for the alternate mail action.**
See "Substituting an Alternate Mail Application" on page 180."

3. **Stop** `sendmail`**.**

4. **Bring up the Admin Edit action from the System_Admin folder.**

5. **Enter** `/usr/dt/appconfig/types/C/dtwm.fp` **file in the File to Edit field.**

6. **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
}
```

   a. **Leave TYPE, CONTAINER_NAME, CONTAINER_TYPE, and POSITION_HINTS as shown below.**

```
TYPE             icon
CONTAINER_NAME   Top
CONTAINER_TYPE   BOX
POSITION_HINTS   5
```

**b. Change the ICON field to identify the icon of the replacement application.**

```
ICON              OWmailtool
```

**c. Change the LABEL field to change the icon label that appears with the icon of the replacement application in the mail subpanel.**

```
LABEL             OW Mail Tool
```

**d. Leave ALTERNATE_ICON and MONITOR_TYPE as shown below.**

```
ALTERNATE_ICON    DtMnew
MONITOR_TYPE      mail
```

**e. Change DROP_ACTION or leave as shown below.**

```
DROP_ACTION       Compose
```

Other mailers may or may not have a Compose action. 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.

**f. Change the PUSH_ACTION field to identify the replacement action to be run when the user clicks on the new mail icon.**

```
PUSH_ACTION       OWmailtool
```

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/C` directory.

**g. Leave the PUSH_RECALL action as shown.**

```
PUSH_RECALL       true
```

When `true`, if an application is launched for a second time, a new application is not launched if the icon for the application window is concealed on the workspace. Instead the application window is brought forward.

**h. Change the CLIENT_NAME field to identify the executable for the replacement application.**

```
CLIENT_NAME       mailtool
```

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

**i. Leave the HELP_* entries as is.**

```
HELP_TOPIC        FPOnItemMail
HELP_VOLUME       FPanel
```

**7. Using the File menu options Save the changes and Close the file.**

---

**Note –** The next step is only necessary if you do this procedure after the system is running.

---

**8. Remove all contents of the $HOME/`.dt/fp.dynamics` directory.**

**9. Restart the Workspace Manager from the workspace menu to see the changes to** the `dtwm.fp` **go into effect in the front panel.**

**10. Restart** `sendmail`**.**

### ▼ To Create a Multilevel Action for the Alternate Mail Application

1. **Assume the security administrator role and make sure you are in an ADMIN_LOW workspace.**
2. **Use the** `Admin Edit` **action to bring up the** `/usr/dt/appconfig/types/C/dtwm.fp` **file to edit.**
3. **Find the control section for mail shown below.**

```
CONTROL OW_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. **C*opy* the control text to a file whose name has the** `.fp` **extension, for example,** `mail.fp`**, and quit the** `dtwm.fp` **file.**
   Create the new file `mail.fp` file in a directory such as `/etc` or `/usr/bin` that is in every user's path.
5. **Bring up the Admin Edit action from the System_Admin folder and open the new** *mail*`.fp` **file for editing.**

**6. Edit the CONTROL OW_Mail section shown below.**

```
CONTROL OW_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
}
```

**a. Leave TYPE, CONTAINER_NAME, CONTAINER_TYPE, and POSITION_HINTS as shown below.**

```
TYPE             icon
CONTAINER_NAME   Top
CONTAINER_TYPE   BOX
POSITION_HINTS   5
```

**b. Change the ICON field to identify the icon of the replacement application.**

```
ICON             OWmailtool
```

**c. Change the LABEL field to change the icon label that appears with the icon of the replacement application in the mail subpanel.**

```
LABEL            OW Mail Tool
```

**d. Leave ALTERNATE_ICON and MONITOR_TYPE as shown below.**

```
ALTERNATE_ICON     DtMnew
MONITOR_TYPE       mail
```

**e. Change DROP_ACTION or leave as shown below.**

```
DROP_ACTION        Compose
```

Other mailers may or may not have a Compose action. 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.

**f. Change the PUSH_ACTION field to identify the replacement action to be run when the user clicks on the new mail icon.**

```
PUSH_ACTION        OWmailtool
```

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/C` directory.

**g. Leave the PUSH_RECALL action as shown.**

```
PUSH_RECALL        true
```

When `true`, if an application is launched for a second time, a new application is not launched if the icon for the application window is concealed on the workspace. Instead the application window is brought forward.

**h. Change the CLIENT_NAME field to identify the executable for the replacement application.**

```
CLIENT_NAME        mailtool
```

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

i. **Leave the HELP_* entries as is.**

```
HELP_TOPIC          FPOnItemMail
HELP_VOLUME         FPanel
```

7. **Save the changes and quit the file.**
8. **Give users the procedure "To Install an Alternate Mailer in the Front Panel" on page 188, after testing the procedure yourself to see if the mailer shows up and works properly.**

## ▼ To Install an Alternate Mailer in the Front Panel

1. **Obtain the pathname of the correct alternate mail application's control file from the system administrator.**
   The system administrator must have done some setup for the mailer to watch for and notify you about mail at all labels within your personal accreditation range. See "To Create a Multilevel Action for the Alternate Mail Application" on page 185.

---

**Note –** Do not install an alternate mailer if the file does not end with a suffix of `.fp`.

---

**Caution –** Unless you have the approval of your site's security administrator, do not install an alternate mailer from any of the application manager folders or if the file does not end with a suffix of `.fp`.

---

2. **Ask the security administrator to stop `sendmail`.**
3. **Using the File Manager, change to the directory where the alternate mail application's control file resides.**
4. **Click the Mailer subpanel access button to bring up the subpanel.**

5. **Drag the icon for the alternate mailer's front panel control file onto the Install Icon dropsite in the Mail subpanel.**
   The icon for the alternate mail application should appear in the Mail slider.
6. **Click the right mouse button while the pointer is over the alternate mail and select Copy to Main Panel.**
7. **For each of the old mail icons in the subpanel, click the right mouse button while the pointer is over any of the mail icons for the old application and select Delete.**
   Repeat this until all of the old icons have been removed. You cannot have a mixture of mail applications running at the same time.
8. **Select Restart Workspace Manager from the Workspace Menu.**
   The size of the subpanel does not adjust correctly until the Window Manager is restarted.
9. **Restart** `sendmail`**.**

# 6

# *User Manager Data Collection Worksheet*

**Note –** In the User and Role Account Worksheet on the following page, the names in the left column correspond to the labels on the buttons on the User Manager Navigator.

# 7

## *User or Role Account Worksheet*

| | |
|---|---|
| **Full Name and Function of Individual or Role** | |
| **Primary Host Name** | |
| **Domain Name** | |

| | |
|---|---|
| **Identity** | Login name: UID: Primary GID: Secondary GIDs: |
| | Comment: |
| | Shell: [ ] Bourne [ ] C [ ] Profile [ ] Other: |
| | User Type: [ ] Normal [ ] Administrative Role [ ] Non-administrative Role |
| **Home** | Create home dir automatically? |
| | Home directory path: |
| | Path to setup files: |
| | Default permissions: |
| | Mail server: |
| **Password** | [ ] Cleared until first login [ ] Account is locked [ ] No password (setuid only) |
| | [ ] Normal Password (Manual [ ] Normal Password (Automatic)<br>NOTE: Normal password options above immediately prompt the administrator to create a password for the account. |
| | Minimum days between password change: |
| | Maximum days before change must be made: |
| | Expiration date: Warning (days before expiration date): |
| | Generation (method used by account when changing password): [ ] Automatic [ ] Manual |
| | Account state: |
| | NIS+ credentials? |
| **Idle** | Idle time : [ ] Minutes before idle action [ ] Forever [ ] System Default |
| | Idle action: [ ] Logout [ ] Lock Screen |
| **Labels** | Clearance: |
| | Minimum sensitivity label: |
| | View : [ ] Internal (replace names of administrative labels with closest equivalents within the user accreditation range) [ ] External (show actual administrative labels) [ ] SysDefault |
| | Sensitivity Labels: [ ] Show [ ] Hide; Information Labels: [ ] Show [ ] Hide |
| | Information Labels: [ ] Show [ ] Hide |
| **Profiles** | See Appendix B, “Profile Definition Tables” for names of profiles and the tools they contain. |
| **Roles** | [ ] Security Administrator [ ] System Administrator [ ] Operator [ ] Root |

# *Managing Execution Profiles for Users and Roles*

*8*

This chapter describes how to modify or add execution profiles, which are used for defining the capabilities of individual users and administrative roles. It includes the following major topics.


| | |
|---|---|
| *Review of Terms* | *page 194* |
| *Background on Execution Profiles* | *page 194* |
| *Use of the Profile Manager to Create or Modify Execution Profiles* | *page 198* |
| *Picking a Naming Service* | *page 199* |
| *Filtering Profiles* | *page 200* |
| *Bringing Up a Blank Profile Definition, Loading an Existing Profile, or Saving Changes Within the Profile Manager* | *page 207* |
| *Entering or Changing the Profile Name or Description* | *page 208* |
| *Switching Among Actions, Commands, and Authorizations Modes* | *page 209* |
| *Working With the Excluded and Included Lists* | *page 209* |
| *Working With Common Features of the Commands and Actions Modes* | *page 211* |
| *Working in Command Mode* | *page 217* |
| *Working in Authorizations Mode* | *page 219* |
| *Working in Actions Mode* | *page 221* |
| *Specifying a New Profile* | *page 227* |
| *Modifying an Existing Profile* | *page 228* |

# 8

This chapter includes the following procedures:


| | |
|---|---|
| *To Access the Profile Manager* | *page 223* |
| *To Pick a Naming Service and Filter for Profiles* | *page 224* |
| *To Enter the Name and Description for a New Profile* | *page 229* |
| *To Specify Commands in the Profile Manager* | *page 229* |
| *To Specify Actions in an Execution Profile* | *page 230* |
| *To Specify Commands in the Profile Manager* | *page 229* |
| *To Specify Authorizations in an Execution Profile* | *page 232* |


## Review of Terms

This section reviews some Trusted Solaris concepts and terms related to execution profiles and introduces some new ones. The reviewed topics were introduced in the *Trusted Solaris User's Guide* and discussed in the *Trusted Solaris Administration Overview* and are included here for convenience.

### Execution profiles

Execution profiles are a bundling mechanism for defining the capabilities of individual users and administrative roles. Each execution profile potentially contains a list of UNIX commands, CDE actions, and authorizations, and the execution profile may also associate security attributes (an *effective UID* and *effective GID*, *inheritable privileges*, or *label constraints*) with each command and action. One or more execution profiles are assigned to each user and role. The order in which execution profiles are specified is significant. When multiple profiles are assigned to a user or role, the set of authorizations is combined, and whatever command or action appears first in the list of profiles for that account is used with the privileges, UID, and GID as specified for its first appearance. Execution profiles can be used to restrict users to a limited set of applications or commands. They also can be used to extend the capabilities of an account, especially when used in combination with the role mechanism. See *enabling attributes* and *restrictive attributes.*

## *Effective UID and GID*

A process executing a command has an effective user ID (EUID) and effective group ID (EGID) equal to the process' real UID and real GID, unless the process has executed a command that has the set user ID (setuid) bit or set group ID (setgid) bit set. Setting an effective UID, an effective GID, or both an effective UID and GID, by means of the setuid or setgid bit is usually done when a command performs checks while it is running to ensure that it is being run by root (UID 0). Setting the setuid or setgid bit on a command or action allows a non-administrative user to do some things that only the superuser could do in the base system. To achieve the same effect in the Trusted Solaris system, the security administrator role can assign an effective UID or effective GID to a command or action in an execution profile when the command or action must be run by a specific user or group, most often when the command must be run as root.

## *Actions*

Actions are a bundling mechanism that allows one or more commands to be specified for a particular task that in turn may be assigned to one or more users. An action can have a set of options and arguments specified along with each of the command(s) and may make use of a dialog box to prompt for additional arguments.

A set of administrative actions have been created to edit files that are not managed through NIS+. For example, to edit the `vfstab_adjunct` file, the security administrator uses an action called Set Mount Attributes. The Set Mount Attributes action includes a command (`adminvi`), an argument (`vfstab_adjunct`), a label range (ADMIN_LOW to ADMIN_LOW), and an effective UID of root and an effective GID of sys. The label range is used to ensure that the `vfstab_adjunct` file is not edited at any other SL. The effective UID and effective GID ensure access to the file and ensure that the file's owner and group do not get changed in the process of being edited.

Each action usually has its own icon, is assigned its own set of security attributes, and may be specified in an execution profile. The icons in the Application Manager and most of the icons in the front panel correspond to actions.

Adding actions is described in the *CDE User's Guide*, and the *CDE Advanced User's and System Administrator's Guide*, but certain aspects of Trusted Solaris security policy affect who can add actions, and who can use the actions. Normal users can drag and drop actions from the Application Manager to the

front panel to run at a single sensitivity label. The system administrator can register actions and put them in the `/etc/dt/appconfig/types/C` directory to make them available for everybody. See "Actions" on page 441 in Chapter 16, "Adding Software," for more about actions and how to add them.

## *Enabling Attributes*

The optional set of override privileges, effective user ID, and effective group ID that may be associated with commands and actions in execution profiles extends the capabilities of the command or action, enabling the account that invokes it to:

- Work outside of the Trusted Solaris security policy
- In a controlled way
- For a well-defined purpose

The UID, GID, and privileges are referred to as *enabling attributes*.

## *Restrictive Attributes*

The label range is referred to as a *restrictive attribute*, because it limits the range of labels at which the command or action may be invoked. The default label range is the full range in the system accreditation range, from ADMIN_LOW to ADMIN_HIGH.

## *Privileges in Profiles*

When a command or an action is assigned one or more privileges in an execution profile, the privileges become available to the command or action through *privilege inheritance*. Privileges are considered to be *enabling attributes*. Commands inherit privilege only when invoked in a profile shell, `pfsh(1MTSOL)`. Actions inherit privileges through the window system, through the same database lookup used by the profile shell.

Using privilege inheritance allows finer controls on who can make use of privilege. Compare the use of inherited privileges to the use of forced privileges for a command. Forced privileges are associated with an executable file, so forced privileges are in effect for the command no matter who invokes the command and no matter what shell is used. In contrast, inherited

privileges are available only when a user or role executes the command in a profile shell and only when one of that account's execution profiles has the command specified to run with privilege.

## *Background on Execution Profiles*

Adding and modifying profiles is an advanced topic for sites that want to control what users can do and enhance what administrators can do—beyond the controls provided in the default system.

The Trusted Solaris default set of execution profiles includes:

- Basic profiles for normal users
- Three administrative role profiles and one non-administrative role profile

Divided among the default role profiles are all the commands, actions, authorizations, and privileges necessary to fulfill both the administrative responsibilities of the UNIX system administrator (root/superuser) and the added administrative responsibilities of managing a Trusted Solaris system.

The default execution profiles may be all your site ever needs to operate fully and with the full measure of trust.[1] However, once the site's security administrator is thoroughly familiar with the execution profiles and how they work, the security administrator may decide to create new execution profiles, based on the site's answers to the following questions:

- What tasks need to be accomplished system-wide and by each individual?
- What is each individual capable of?
- What information and responsibilities can each individual be trusted with?

Before making any changes to execution profiles, especially before you create any new administrative role profiles or modify the administrative role profiles that currently exist, make sure you thoroughly understand how the current roles are configured—as described in tables in the *Trusted Solaris Administration Overview*—and understand what the security implications might be of any changes or additions you might decide to make. The tables list the commands, actions, and authorizations assigned to each profile, and the security attributes (UID, GID, privileges, and label range) that apply to each command and action.

1. Trusted Solaris documentation sometimes uses the term profile and execution profile interchangeably. The term execution profile is stressed to avoid any confusion with the installation profiles that may be created during installation and used as part of system jumpstart.

# 8

## Use of the Profile Manager to Create or Modify Execution Profiles

In the default configuration, the Profile Manager is assigned only to the security administrator role. The Profile Manager is used at ADMIN_LOW.

When the Profile Manager is launched (as described in "To Access the Profile Manager" on page 223), the Profile Manager Load dialog box displays as shown in Figure 8-1 on page 198. It offers menus for picking a naming service and filtering profiles. Filtering profiles allows you to start with an existing profile loaded into the Profile Manager or to start with the Profile Manager empty.

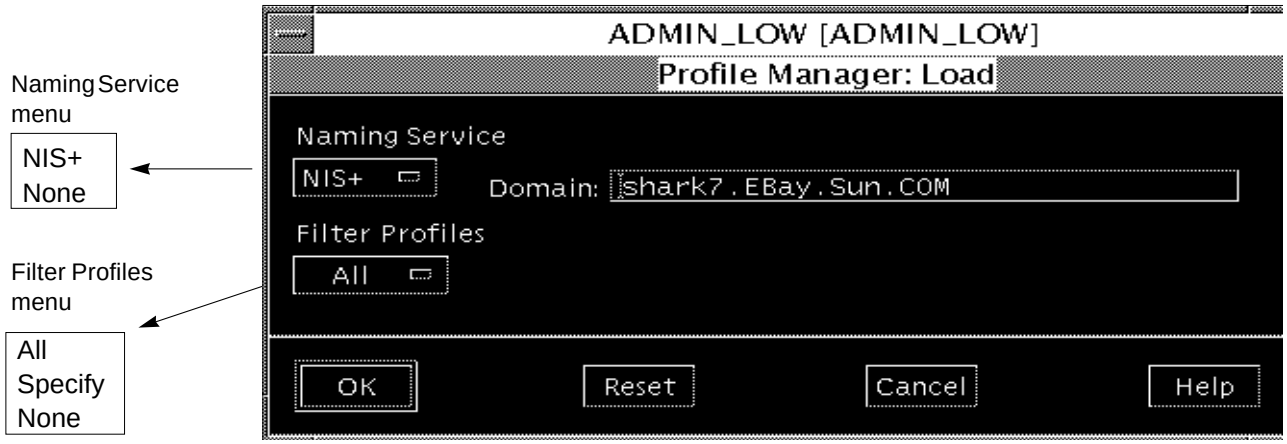



*Figure 8-1* Profile Manager: Load Dialog Box

## Using the Control Buttons on the Profile Manager Dialog Boxes

The control buttons shown in Figure 8-1 are also found on the Set Privileges and Set UID/GID dialog boxes.

### *OK*

OK saves any changes and closes the dialog box.

### *Reset*

Restores the fields to their original state after the last save and leaves the dialog box open.

***Cancel***

Cancel closes the dialog box without saving any changes

***Help***

Help provides context sensitive help messages.

## *Picking a Naming Service*

Like all other Trusted Solaris AdminSuite tools, the Profile Manager requires that you pick a naming service. The default is NIS+, which is the recommended mechanism for centralized distribution of user and host information. The None option should only be used in specialized circumstances where a knowledgeable security administrator decides that local profiles on individual hosts are both needed and allowable within the site's security policy. A standalone Trusted Solaris host may use either naming service.

If you choose NIS+ for the Naming Service, any modifications you make to the set of execution profiles are stored in the NIS+ `tsolprof` table for the specified domain when the Profile Manager Load dialog first displays. The current domain name appears in a text field next to the NIS+ option, as shown in Figure 8-2. If the secadmin role has been authenticated for another domain, you may update the NIS+ profiles table in that other domain by replacing the domain name with that of the other NIS+ domain.

*Figure 8-2* Profile Manager: Load, Naming Service NIS+

If you choose None for the Naming Service, the name of the local host displays as shown in Figure 8-3. You may change the name in the Host field. Any modifications you make to the set of execution profiles are stored in the `/etc/security/tsol/tsolprof` file on the specified host.

*Figure 8-3* Profile Manager: Load, Naming Service None

### *Decision To Make Before Starting*

- **Decide whether to use NIS+ or no naming service.**

## *Filtering Profiles*

The Filter Profiles menu shown in Figure 8-4 on page 200 allows you to specify whether the Profile Manager should come up empty or come up loaded with a profile that you specify. Which of the Filter Profile menu options you choose depends on whether you are adding a new profile or modifying an existing profile. See “When Adding a New Profile” or “When Modifying an Existing Profile” on page 201.

**Note –** Whichever option you choose to start, after the Profile Manager comes up you may display another profile or empty the display by using the Profiles menu, as described in “Bringing Up a Blank Profile Definition, Loading an Existing Profile, or Saving Changes Within the Profile Manager” on page 207.

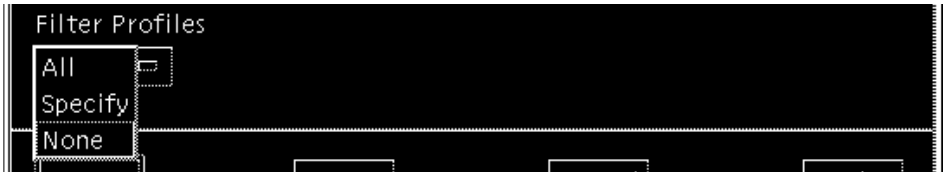



*Figure 8-4* Profile Manager: Load, Profile Filter Choices

**Note –** The Profile Manager initially displays in *action mode*, with lists of excluded and included actions. (Figure 8-6 on page 202 shows one example of the Profile Manager in action mode.) You can use the Profile Manager View menu to switch among the modes for actions, commands, and authorizations, as described “To Specify Commands in the Profile Manager” on page 229.

***Decision To Make***

- **Decide whether you are adding a new profile and go to the appropriate section, either “When Adding a New Profile” or “When Modifying an Existing Profile” on page 201.**

## *When Adding a New Profile*

If you are adding a new profile, you can bring up the Profile Manager either of these two ways:

- Empty (as described under “Launching an Empty Profile Manager” on page 201)
- Loaded with an existing profile that you may then rename and modify (as described under “Launching the Profile Manager Loaded With an Existing Profile” on page 202).

***Decision To Make Before Starting***

- **Decide whether to copy and modify an existing profile or start with an empty Profile Manager.**

## *When Modifying an Existing Profile*

To modify an existing execution profile, you launch the Profile Manager loaded with the profile (as described under “Launching the Profile Manager Loaded With an Existing Profile” on page 202) and you then make modifications, keeping the same name.

## *Launching an Empty Profile Manager*

Choosing None from the Filter Profiles menu (as shown in Figure 8-5) brings up an empty Profile Manager (as shown in Figure 8-6 on page 202).

*Figure 8-5* Choosing None from the Profile Manager: Load, Profile Filter Menu

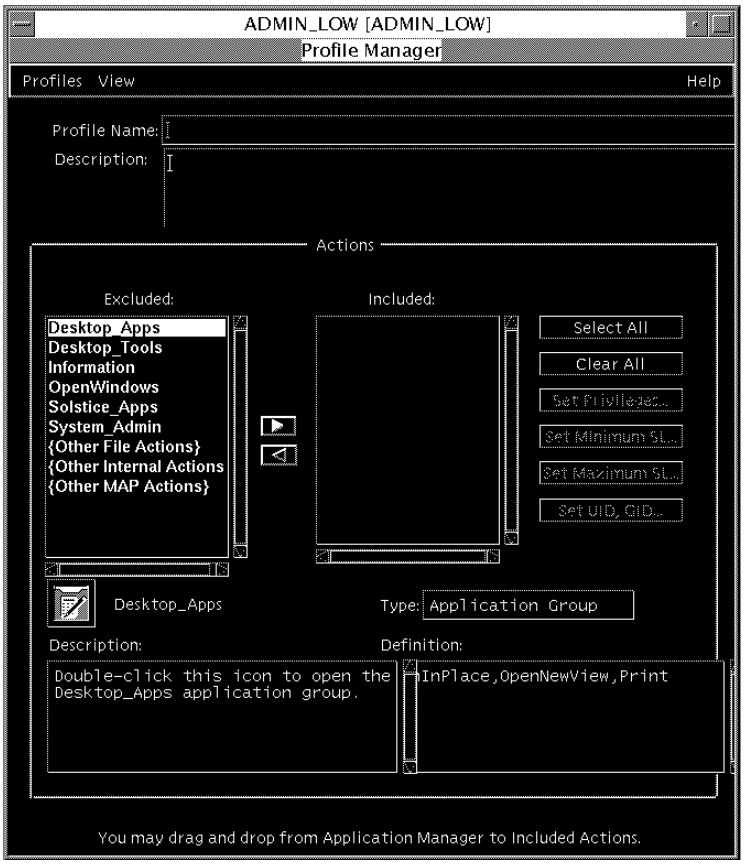



*Figure 8-6* Empty Profile Manager in Action Mode

## *Launching the Profile Manager Loaded With an Existing Profile*

Choosing either All or Specify from the Filter Profiles menu brings up a list from which you choose the name of a profile. When you click Load after selecting the name, the Profile Manager comes up with an existing profile loaded.

- If you want to select an existing profile name from the names of all existing profiles, see list item 1 on page 203.
- If you want to use a regular expression to search for the name of an existing profile, see list item 2.b on page 205.

- If you want to specify the name of an existing profile, see list item 2.a on page 204.

1. Choosing All from the Filter Profiles menu (as shown in Figure 8-7) allows you to pick the name of a profile from a list of all profiles (as shown in Figure 8-8).

*Figure 8-7* Choosing All from the Profile Manager: Load, Filter Profiles Menu

Highlighting the name of the profile and clicking Load (as shown in Figure 8-8) opens the Profile Manager with the profile loaded (as shown in Figure 8-12 on page 206).

*Figure 8-8* Profile Manager: Load, Highlighting a Profile Name

*Figure 8-9* Profile Manager With A Profile Loaded

2. Choosing Specify from the Filter Profiles menu (as shown in Figure 8-10 on page 205) allows you to specify a profile name in a text field in one of the following two methods.

   a. Entering a regular expression (as shown in Figure 8-10) creates a list of profiles to select among.

*Figure 8-10* Specifying a Profile to be Loaded in the Profile Manager By Using a Regular Expression

The example in Figure 8-10 shows P* entered to locate any profile beginning with P. Figure 8-11 shows the result: the Privileged Shells profile displays in the Profile Manager: Open dialog. Highlighting the name of the profile and clicking Load opens the Profile Manager with the Privileged Shell profile loaded (as shown in Figure 8-12 on page 206).

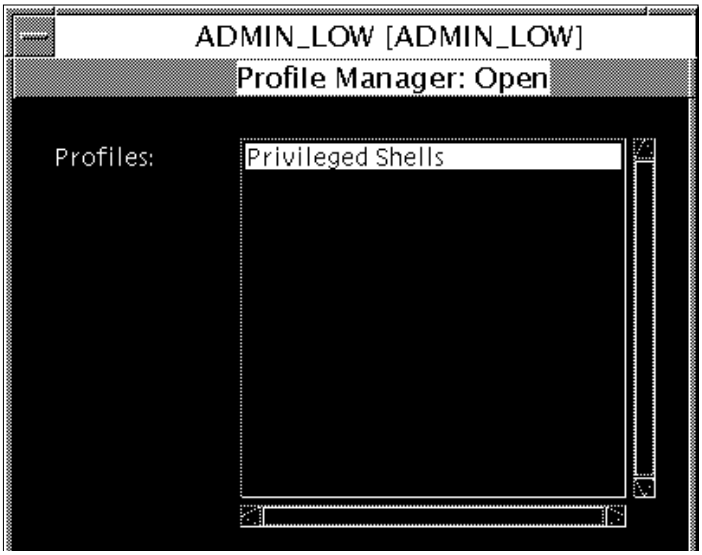



*Figure 8-11* Privileged Shells Profile Listed in the Profile Manager: Open Dialog When P* is Specified in the Filter Profiles Text Field

b. Entering the name of a known profile causes the profile name to be displayed for your selection.

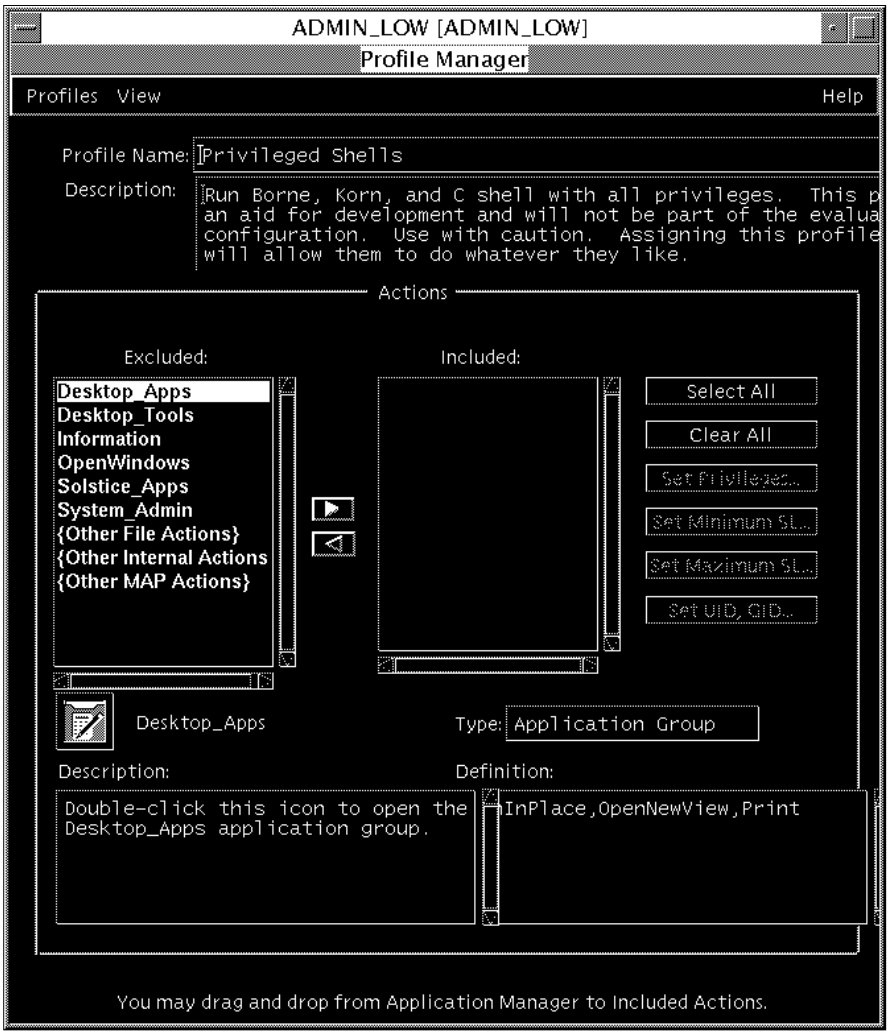



*Figure 8-12* Profile Manager Loaded With the Privileged Shells Profile

## *Bringing Up a Blank Profile Definition, Loading an Existing Profile, or Saving Changes Within the Profile Manager*

Figure 8-13 shows the options on the Profile menu.

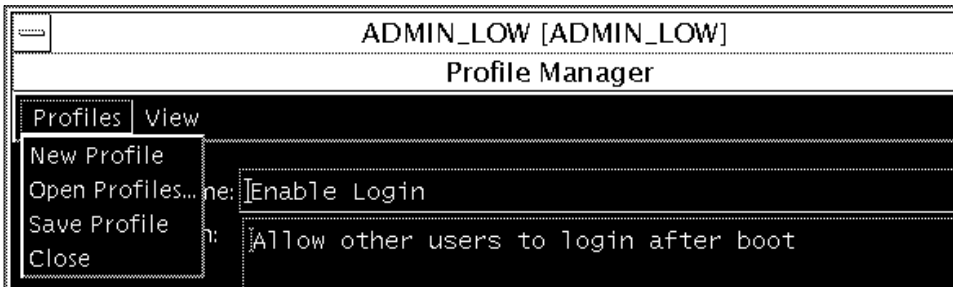



*Figure 8-13* The Profile Manager Profiles Menu For Opening, Saving, and Closing Profiles

You can use the options in the Profiles menu to do the following:

- New Profile clears the existing profile description from the Profile Manager.
- Open Profile allows you to select another profile to load from the list of all profiles. (A Profile Manager: Open dialog displays with a list of all profiles for you to choose from, as shown in Figure 8-14.)
- Save Profile saves any changes you have made to the profile.
- Close closes the Profile Manager without prompting you or saving any unsaved changes.

*Figure 8-14* Profile Manager: Open, Highlighting a Profile Name

## *Entering or Changing the Profile Name or Description*

If you brought up the Profile Manager without a profile loaded, you may enter a new profile name and description in the empty text entry fields. If you enter a name and description and use Profiles->Save, a new profile is created with the specified name.

When the Profile Manager is loaded with an execution profile, you may modify the existing name and description. If you modify an existing profile's name and use Profiles->Save, a new profile is created with the modified name.

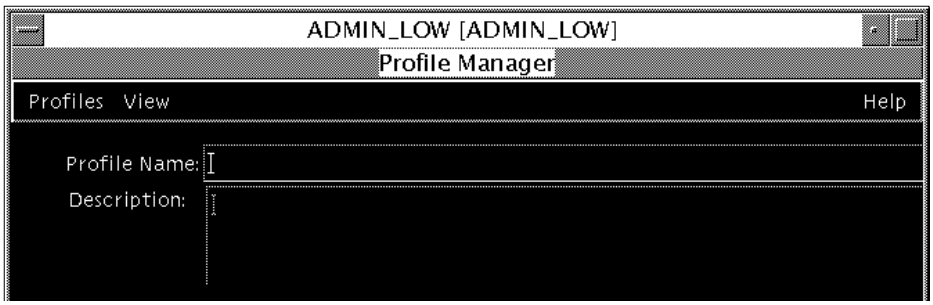



*Figure 8-15* The Profile Name and Description Fields in the Profile Manager

## Switching Among Actions, Commands, and Authorizations Modes

The Actions mode is the first one that comes up after you click OK on the Profile Manager: Load dialog (as shown previously in Figure 8-12 on page 206). Figure 8-16 shows the View menu for switching among the actions, commands, and authorizations modes.

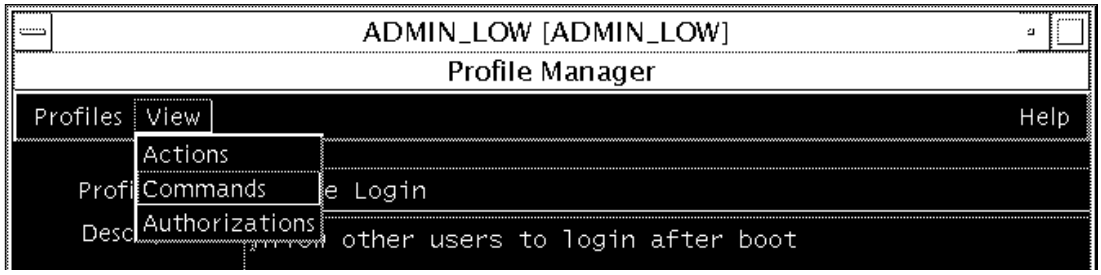



*Figure 8-16* The Profile Manager View Menu For Switching Between Actions, Commands, and Authorizations

See “Working in Command Mode” on page 217, “Working in Authorizations Mode” on page 219, and “Working in Actions Mode” on page 221.

## Working With the Excluded and Included Lists

Figure 8-17 shows an example of the Profile Manager Excluded and Included Lists, which are used for actions, command, authorizations, and privileges. The example shows actions, but the methods for working with the items in these lists apply to any of the Profile Manager dialogs that show excluded and included lists.

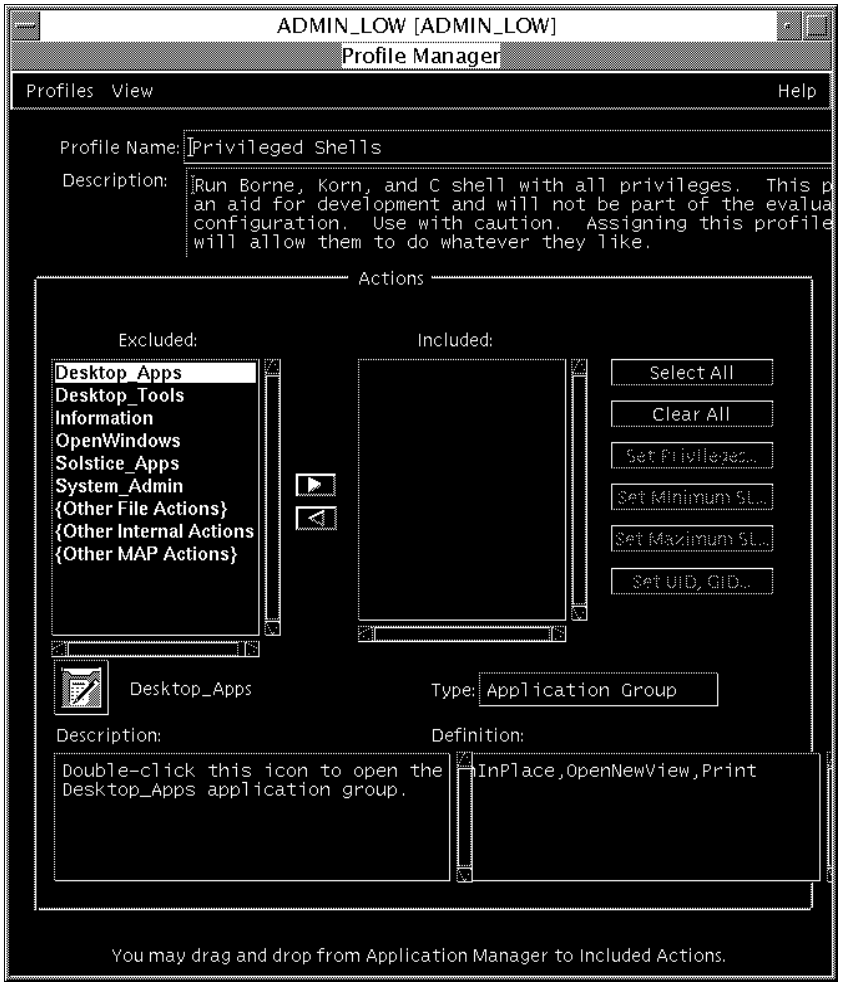



*Figure 8-17* Profile Manager Loaded With the Privileged Shells Profile

## *Moving Items Between Lists*

You can move an item (whether it is an authorization, a privilege, a command, a complete directory, an individual action, or a group of actions in an application group) by highlighting the item's name and using the arrow button to move it to the desired list.

### *Dragging and Dropping Into the Included List*

As noted at the bottom of the Profile Manager, you may drag and drop items that match the current mode from the File Manager into the included list. For example, in commands mode you could drag the icon in for the `mount` command's executable file from the `/etc` folder to the included list.

## *Moving and Clearing Many List Items With the Select All and Clear All Buttons*

All of the Profile Manager modes and the Set Privilege Dialog Box have the Select All and Clear All buttons.

***Select All***

Select All moves all of the items in the excluded list to the included list

***Clear All***

Clear All moves any items in the included list to the excluded list

## *Working With Common Features of the Commands and Actions Modes*

This section describes using common features of the Profile Manager's commands and actions modes in the following subsections


| | |
|---|---|
| *Expanding and Contracting Application Group and Directory Listings in the Command and Actions Modes* | *page 211* |
| *Using the Buttons to Set Security Attributes on Commands and Actions* | *page 212* |
| *Setting Privileges on Commands and Actions* | *page 214* |
| *Setting a Label Range for a Command or Action* | *page 215* |


### *Expanding and Contracting Application Group and Directory Listings in the Command and Actions Modes*

Each heading in the default excluded list stands for a group of actions (in the Actions Mode) or commands (in the Commands Mode). By highlighting the heading and double-clicking on it or on its icon (if one exists), you can view all the individual items (actions, or commands) it contains. For example, when

you double-click on the highlighted Solstice Apps action heading shown previously in Figure 8-17 on page 210, it expands to list all the actions in that grouping, as shown in Figure 8-18 on page 212.

By double-clicking on an expanded command heading, you can contract the list back to its heading or directory name. Items can be moved between the excluded and included list one by one or they may be moved in a group by moving the heading or directory name.

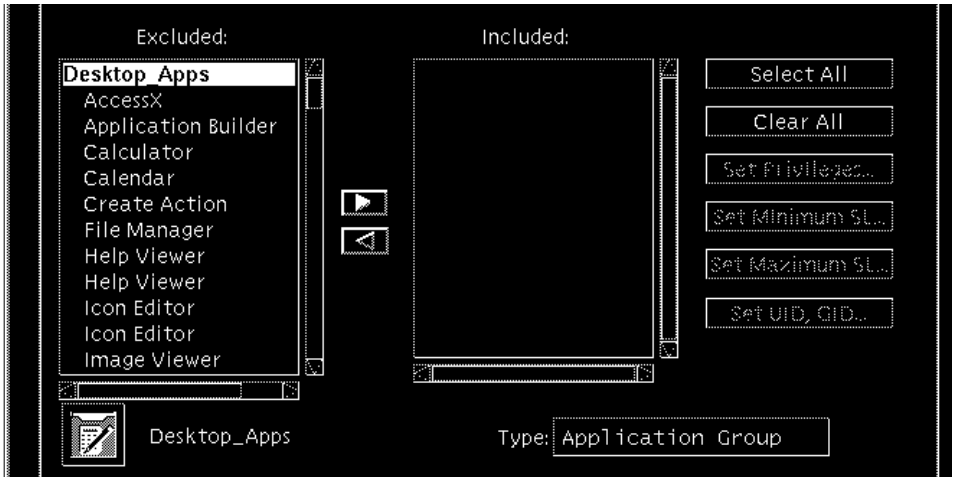



*Figure 8-18* Expanding a Grouping Name to List All of Its Contents

## *Using the Buttons to Set Security Attributes on Commands and Actions*

The buttons shown in Figure 8-20 on page 213 only display when an item in the included list is highlighted and only when security attributes may be set for that item.

*Figure 8-19* Buttons for Setting Privileges, Label Range, UID and GID

Figure 8-20 shows the format command highlighted, and because privileges, an SL range, UID and GID may be set on commands, the buttons are visible and usable. In contrast, because Tool Talk messages cannot have security attributes, when an action that a Tool Talk message (TT_MSG appears in the description field) is highlighted, the buttons are grayed.

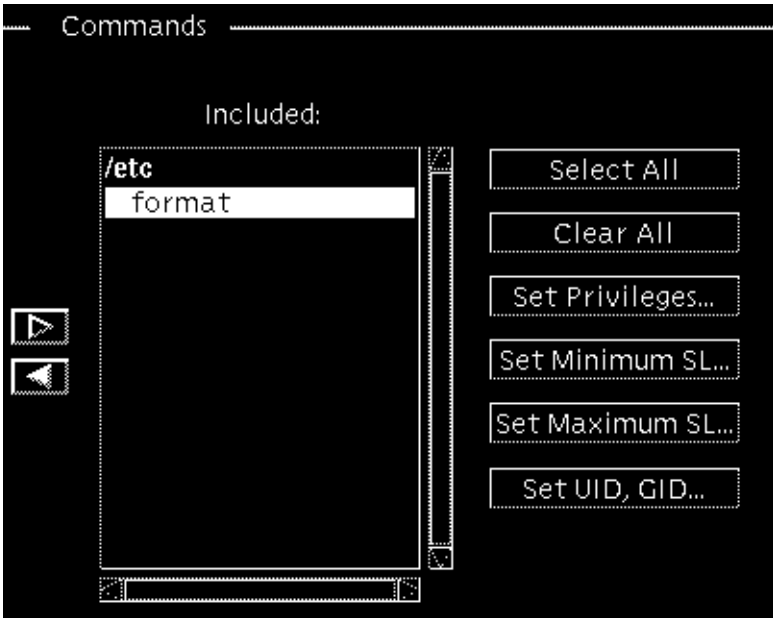



*Figure 8-20* Buttons for Setting Privileges, Label Range, UID and GID

## *Setting Privileges on Commands and Actions*

After a command or action is highlighted in the Profile Manager, clicking the Set Privileges button brings up the Set Privileges dialog, as shown in Figure 8-21. A number of privileges are listed in the excluded list. Whether the included list is empty or not depends on whether the command has previously been assigned any privileges.

### ***Description***

The Description field describes what the privilege allows the process executing the command to do to bypass security policy.

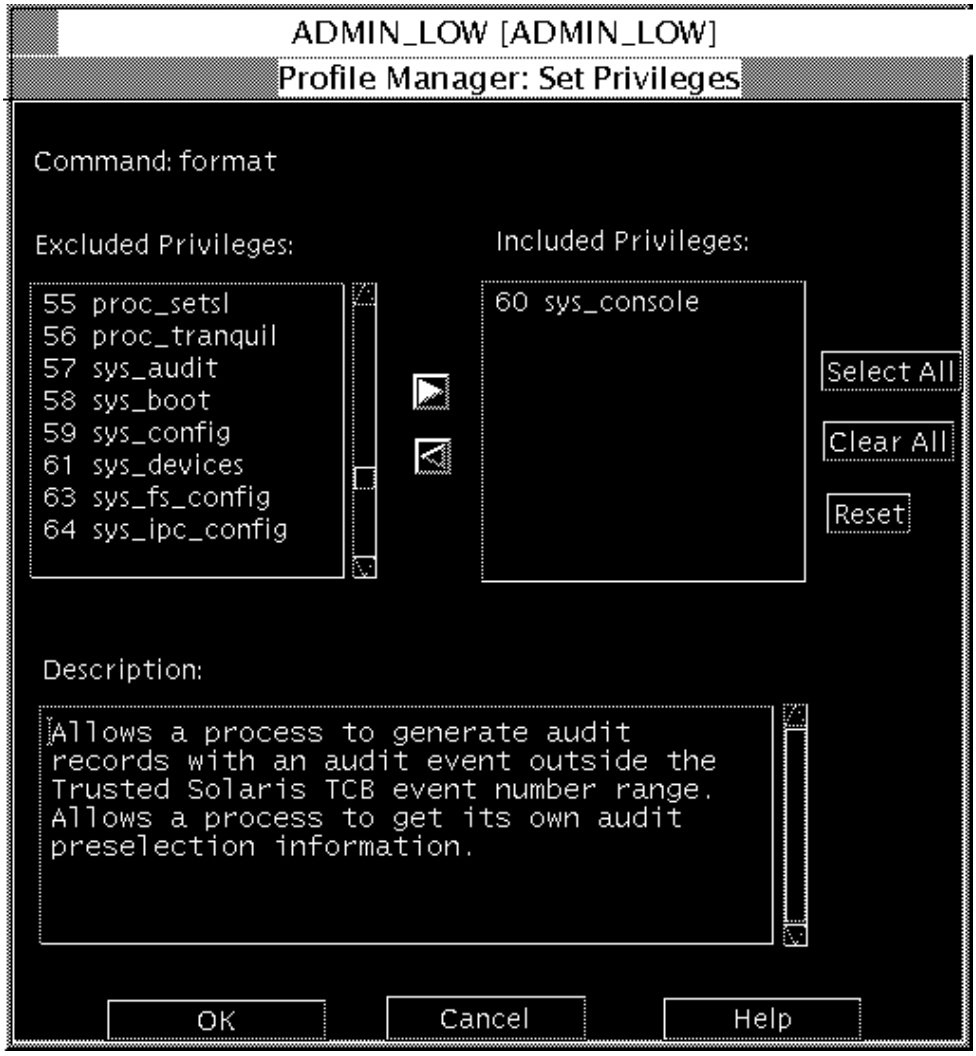



*Figure 8-21* Profile Manager: Set Privileges Dialog Box

## *Setting a Label Range for a Command or Action*

After a command or action is highlighted in the Profile Manager, you set a label range by setting a minimum and maximum sensitivity label. Clicking the Set Minimum SL or Set Maximum SL button brings up a label builder. (The Set

Minimum SL dialog is shown in Figure 8-21.) Using these dialogs to set the minimum and maximum SL of a command or action is the same as using any other Trusted Solaris label builder.

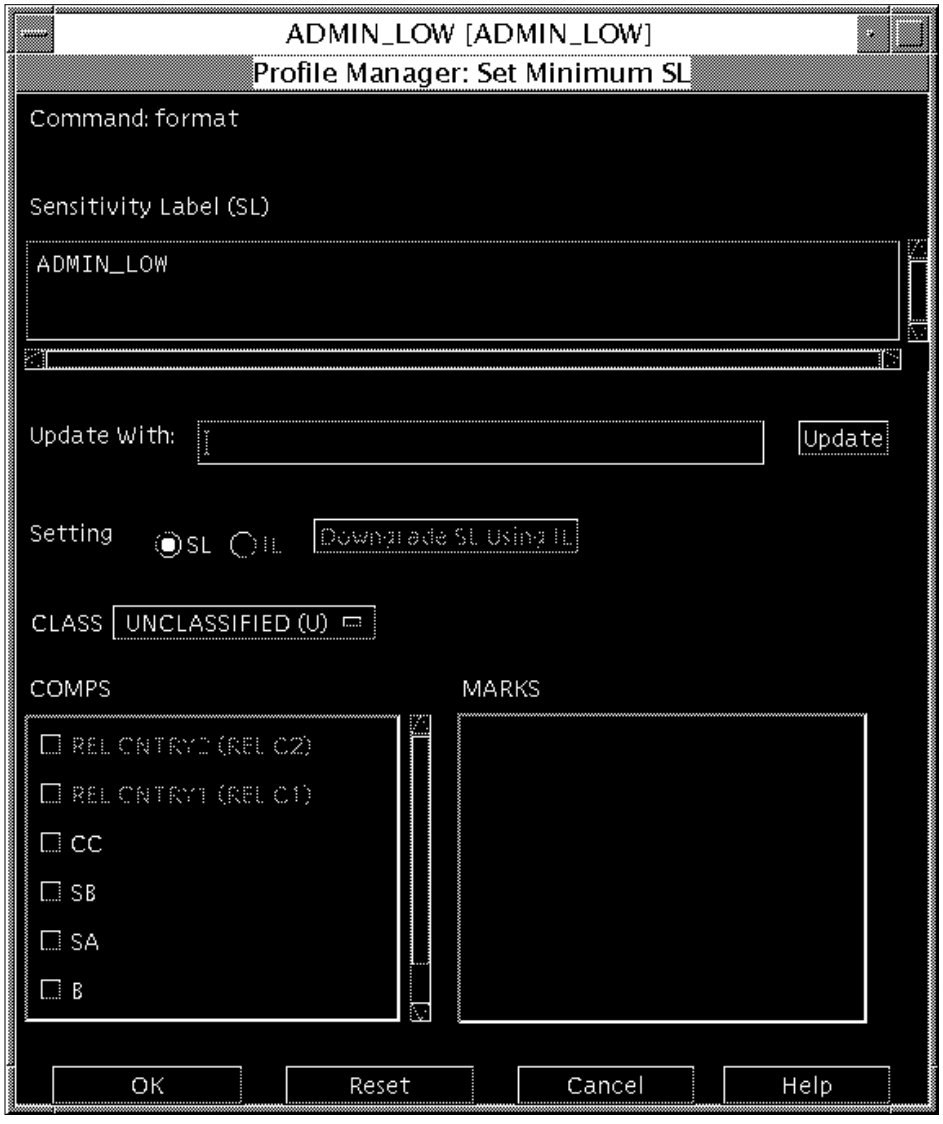



*Figure 8-22* Profile Manager: Set Minimum SL Dialog

## *Working in Command Mode*

This section explains how to use features of the command mode that are not on any of the dialogs in the authorizations and actions modes. It does not duplicate information provided in the following sections on topics that apply to working in command mode:


| | |
|---|---|
| *Entering or Changing the Profile Name or Description* | *page 208* |
| *Working With the Excluded and Included Lists* | *page 209* |
| *Working With Common Features of the Commands and Actions Modes* | *page 211* |


Selecting Commands from the Profile Manager View menu brings up the command mode, as shown in Figure 8-23 on page 218. A number of directories are listed in the excluded list. Whether the included list is empty or not depends on whether the profile has been assigned any commands.

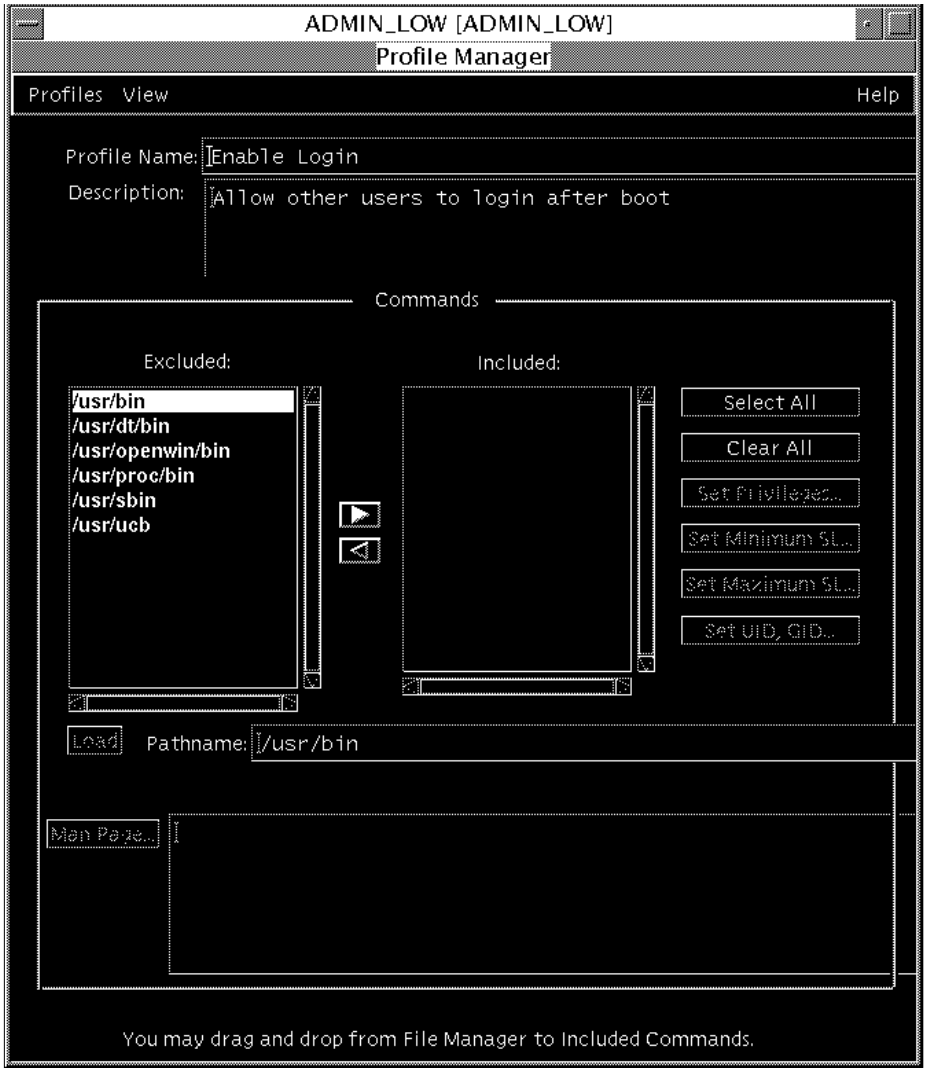



*Figure 8-23* The Profile Manager Command Mode

## *Loading A New Directory*

You may add another directory to the excluded list by entering the pathname of the directory into the text entry field next to the Load button, and then clicking the

load button. In Figure 8-24, the `/etc` directory has been entered into the Pathname: field next to the Load button and loaded into the excluded list.

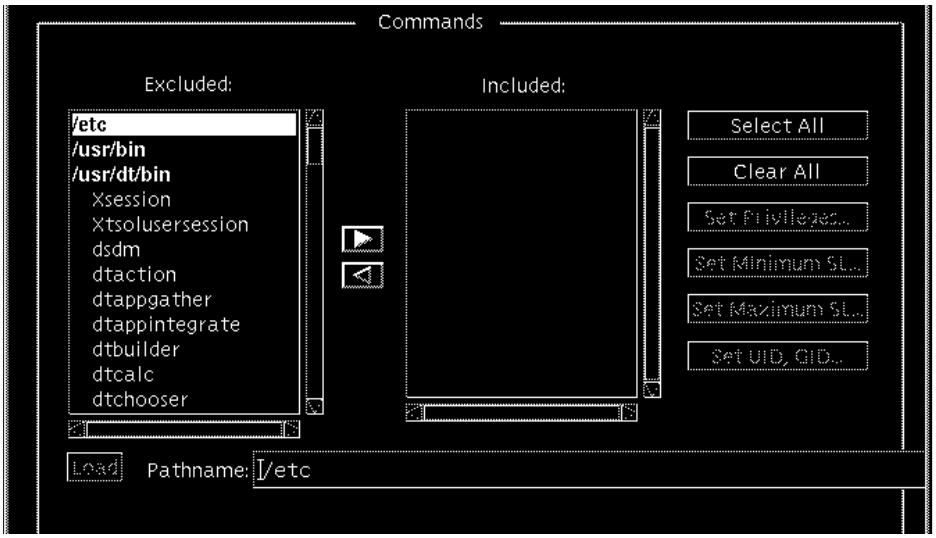



*Figure 8-24* Entering the Pathname of the `/etc` Directory to Choose From Its Commands

### *Viewing a Command's Man Page*

When a command is highlighted in the included list, you can view its man page starting with the DESCRIPTION section by clicking the Man Page button.

## *Working in Authorizations Mode*

The authorizations mode does not have any features that are not on the dialog boxes for the commands and actions modes. Working in authorizations mode is described in the previous sections:


| | |
|---|---|
| *Entering or Changing the Profile Name or Description* | *page 208* |
| *Working With the Excluded and Included Lists* | *page 209* |
| *Working With Common Features of the Commands and Actions Modes* | *page 211* |

Selecting Authorizations from the Profile Manager View menu brings up the authorizations mode, as shown in Figure 8-25. A number of authorizations are listed in the excluded list. Whether the included list is empty or not depends on whether the profile has been assigned any authorizations.

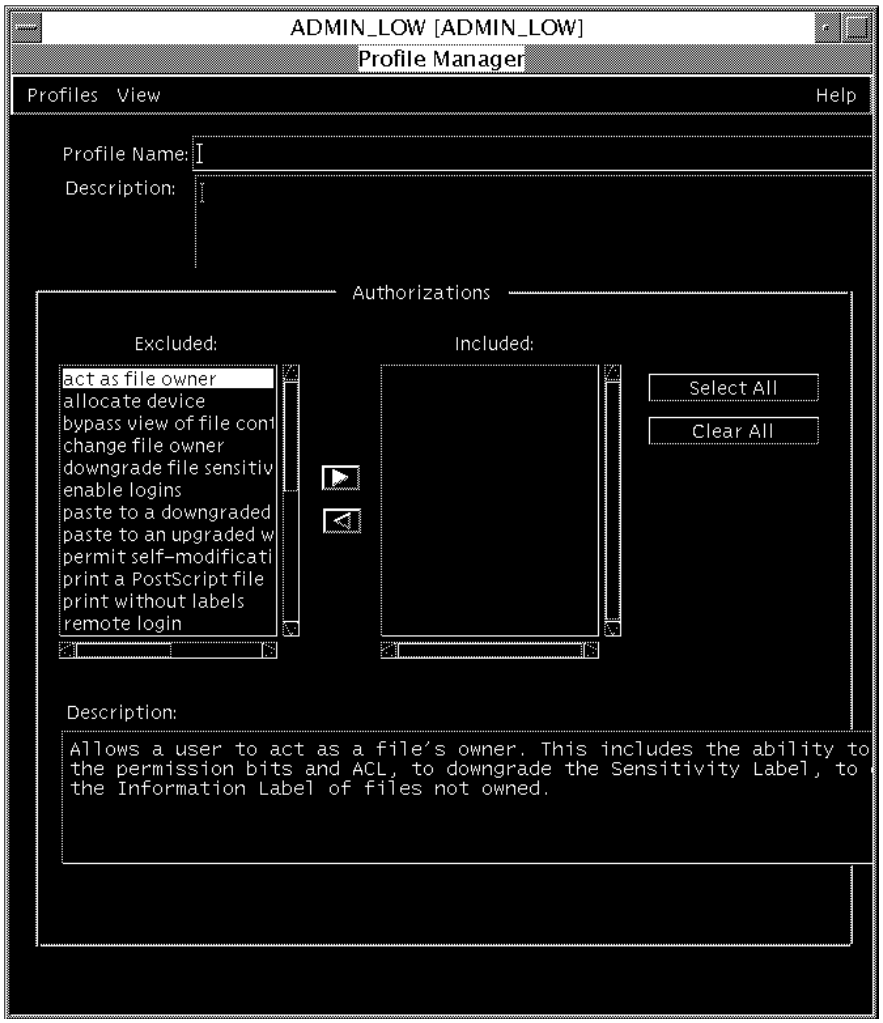



*Figure 8-25* The Profile Manager in Authorization Mode

## Working in Actions Mode

This section explains how to use features of the actions mode that are not on any of the dialogs in the commands and authorizations modes. It does not duplicate information provided in the following sections on these topics that apply to working in actions mode:


| | |
|---|---|
| *Entering or Changing the Profile Name or Description* | *page 208* |
| *Working With the Excluded and Included Lists* | *page 209* |
| *Working With Common Features of the Commands and Actions Modes* | *page 211* |


Selecting Actions from the Profile Manager View menu brings up the actions mode, as shown in Figure 8-27 on page 222. A number of actions are listed in the excluded list. Whether the included list is empty or not depends on whether the profile has been assigned any actions.

### *Icon*

The icon shown in Figure 8-26 is the icon assigned to the highlighted action or application group. As stated in the description section at the bottom of Figure 8-27 on page 222, you can double-click the icon of an applications group to list all of the items contained in the group.

### *Type*

The Type field indicates the type of item highlighted. An application group is a heading that includes a number of actions, which may be expanded to list all its included actions. When an action is highlighted, the possible types are:

COMMAND
TT_MSG

*Figure 8-26* Icon and Type in Action Mode

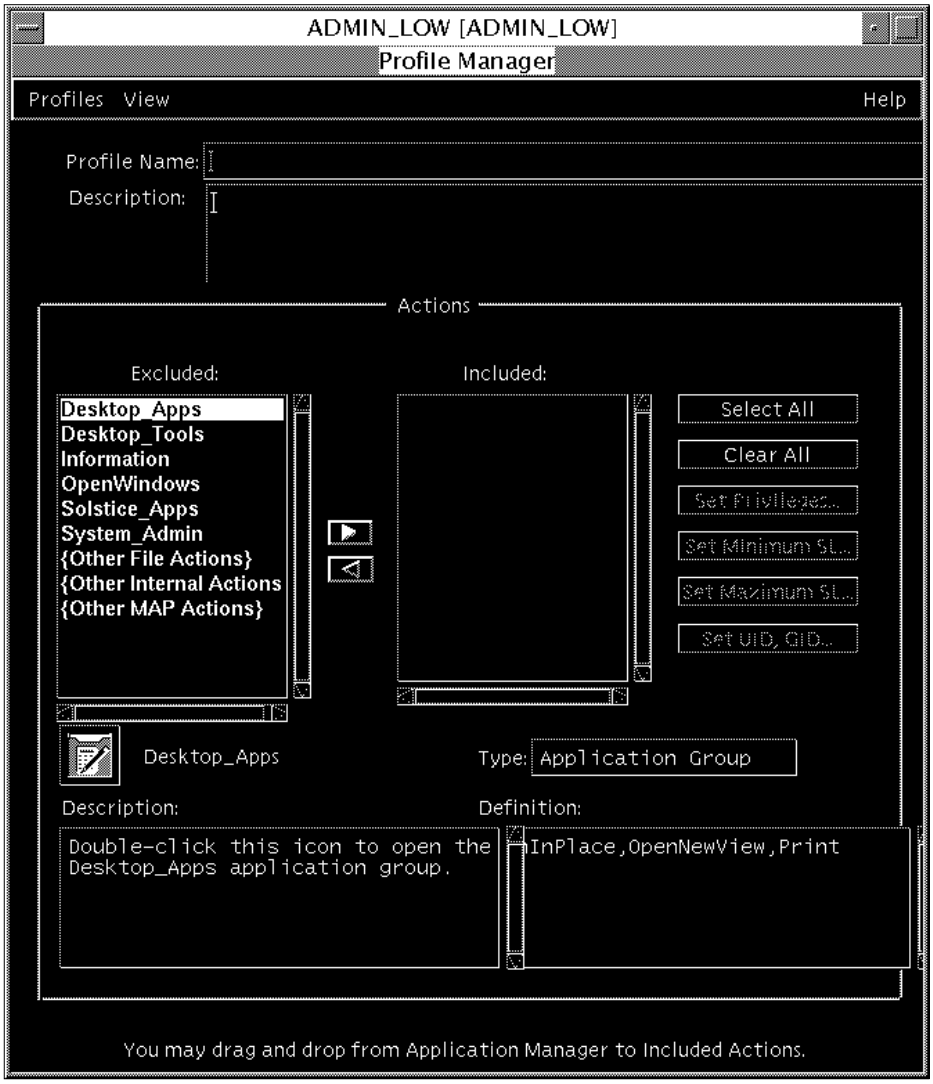



*Figure 8-27* Profile Manager in Action Mode

### ▼ To Access the Profile Manager

1. **Assume the security administrator role.**
2. **In a workspace at ADMIN_LOW, bring up the Profile Manager in one of the following two ways.**

   a. **Access the Profile Manager by clicking on the Profile Manager icon in the Application Manager.**
   Figure 8-28 shows the Profile Manager highlighted in the Solstice_Apps folder.

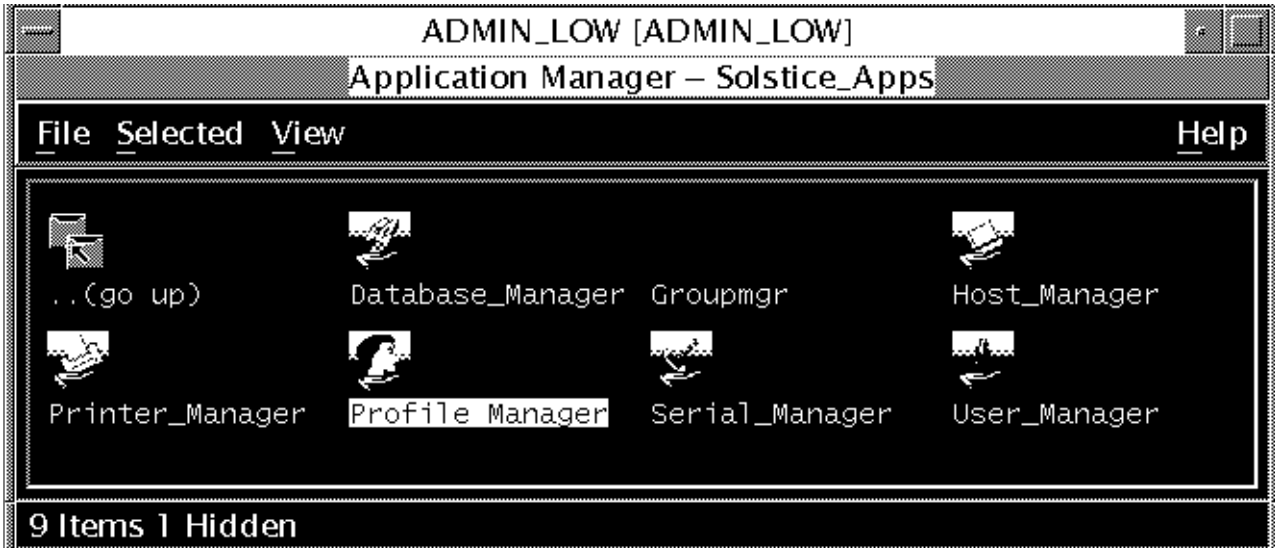



*Figure 8-28* The Profile Manager Icon Highlighted in the Solstice_Apps Folder

   b. **Bring up a terminal running the default profile shell for the security administrator role, and enter** `dtprofile` **on the command line.**

The Profile Manager: Load window displays with the current domain name in the Domain field (see Figure 8-29 on page 224).

### ▼ To Pick a Naming Service and Filter for Profiles

1. **Access the Profile Manager.**
   If needed, see "To Access the Profile Manager" on page 223. The Profile Manager Load Dialog Box displays.

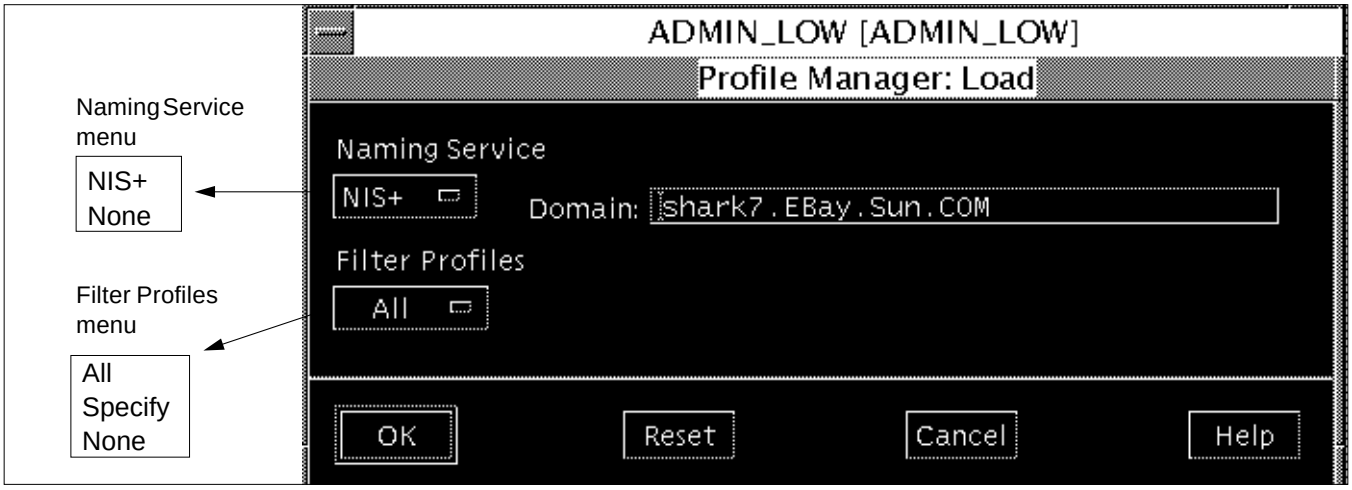



*Figure 8-29* Profile Manager: Load Dialog Box

2. **Choose either NIS+ or None from the Naming Service menu.**

   a. **If you choose NIS+, accept or change the name of the domain in the text field next to the NIS+ menu item (as shown in Figure 8-29).**

   b. **If you choose None for the Naming Service, accept or change the name of the local host in the text field next to the None menu item (as shown in Figure 8-30).**

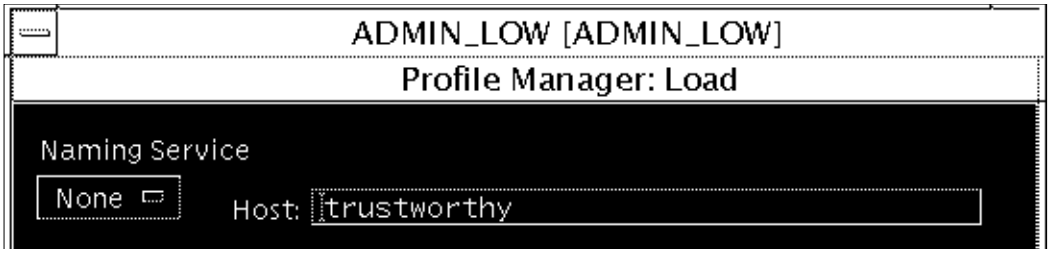



*Figure 8-30* Choosing None from the Profile Manager: Load, Naming Service Menu

3. **Choose an item from the Filter Profiles menu to specify whether you want the Profile Manager to display with all profiles loaded, to display a specified profile, or to display with no profile loaded.**
   Choose All, Specify, or None, as shown in Figure 8-31.

*Figure 8-31* Profile Manager: Load, Profile Filter Choices

   a. **If you choose All, specify a profile from the list of all profiles.**

      i. **Click OK.**

         The Profiles: Load dialog box displays with all the existing execution profiles listed.

      ii. **Select a profile from the list.**

      iii. **Click Load.**

         The Profile Manager displays with the profile or profiles loaded.

   b. **If you choose Specify, specify the name of a profile to load.**
      When you choose Specify, a text entry field displays next to the menu item, as shown in Figure 8-32.

*Figure 8-32* Specifying Profile Names Using a Regular Expression on the Profile Manager: Load, Filter Profiles Menu

i. **Enter the name of an existing execution profile or a regular expression in the text entry field.**
The example in Figure 8-32 shows P* entered to locate any profile beginning with P.

ii. **Click OK or press Return.**
A list of one or more profiles displays in the Profile Manager: Open dialog (as shown in Figure 8-33.

*Figure 8-33* The Privileged Shells Profile Displayed When P* is Specified

iii. **Highlight the desired profile name.**

iv. **Click Load.**

The Profile Manager displays with the profile or profiles loaded.

c. **If you chose None, go to "To Enter the Name and Description for a New Profile" on page 229.**
The Profile Manager displays the list of available actions with no profiles loaded, as shown in Figure 8-34 on page 227.

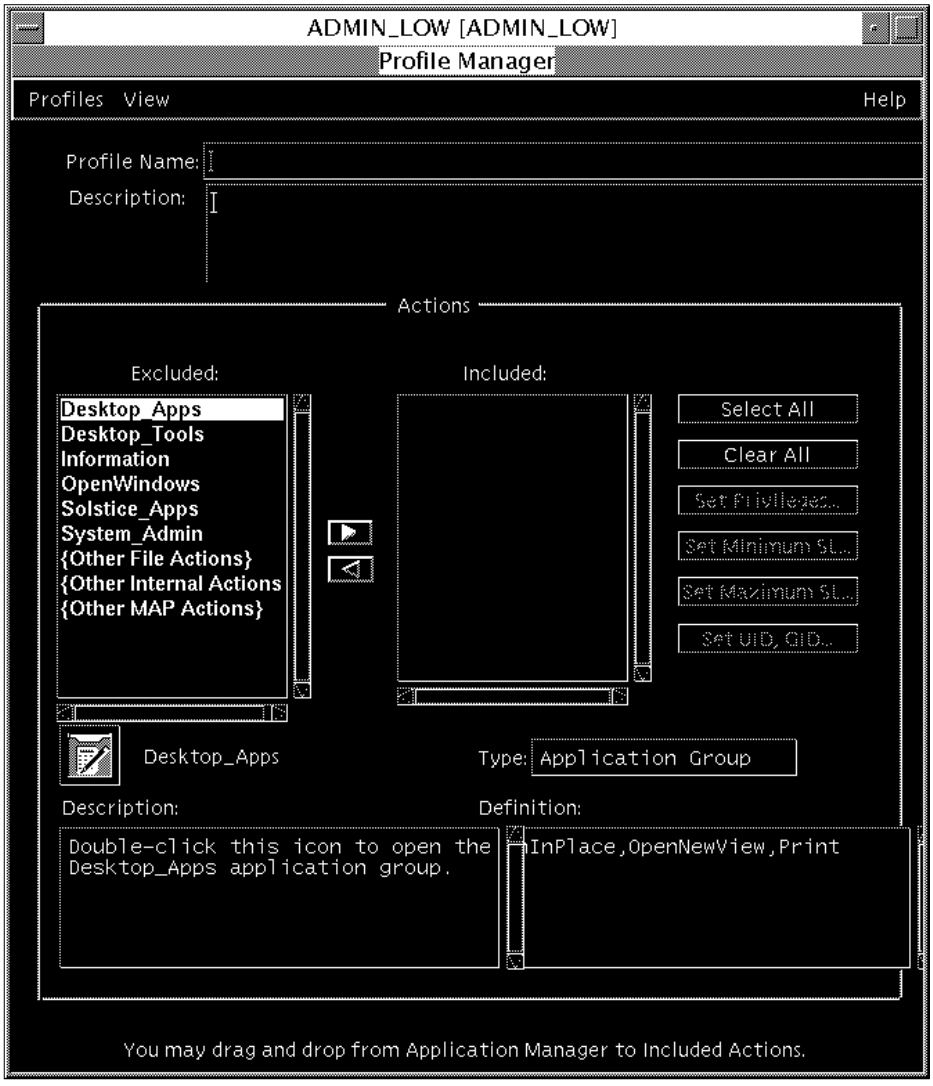



*Figure 8-34* Empty Profile Manager in Action Mode

## *Specifying a New Profile*

Begin with:

| *To Access the Profile Manager* | *page 223* |
|---|---|

When creating a new execution profile, you can rename and modify an existing profile or start with all fields empty.

To specify an existing execution profile, on page 225 under “To Pick a Naming Service and Filter for Profiles),” do Step 3.a or b.

To enter a new name and start with all fields empty, do Step 3.c. on page 226, and go to the following:

| | |
|---|---|
| *To Enter the Name and Description for a New Profile* | *page 229* |

Define the new profiles actions, commands, and authorizations, as desired, as described in:

| | |
|---|---|
| *To Specify Commands in the Profile Manager* | *page 229* |
| *To Specify Actions in an Execution Profile* | *page 230* |
| *To Specify Authorizations in an Execution Profile* | *page 232* |

## *Modifying an Existing Profile*

When modifying an existing profile, start with:

| | |
|---|---|
| *To Access the Profile Manager* | *page 223* |

On page 225 under “To Pick a Naming Service and Filter for Profiles,” do Step 3.a or Step 3.b to specify the name of an existing profile.

Redefine any of the profile’s sets of actions, commands, or authorizations as desired, as described in:

| | |
|---|---|
| *To Specify Commands in the Profile Manager* | *page 229* |
| *To Specify Actions in an Execution Profile* | *page 230* |
| *To Specify Authorizations in an Execution Profile* | *page 232* |

## *Execution Profile Procedures*

### ▼ To Enter the Name and Description for a New Profile

1. **If the Profile Manager is not up, bring it up as described in "To Access the Profile Manager" on page 223 and "To Pick a Naming Service and Filter for Profiles" on page 224.**
2. **Enter the name in the Profile Name: field.**
3. **Enter a description of the profile in the Description: field.**
4. **Choose Save from the Profiles menu.**

### ▼ To Specify Commands in the Profile Manager

1. **If the Profile Manager is not up, bring it up as described in "To Access the Profile Manager" on page 223 and "To Pick a Naming Service and Filter for Profiles" on page 224.**
2. **Choose Commands from the View menu.**
3. **Put the name of the command you want to specify into the included list.**
   If the directory where the command resides is not in the default list, do Step a and Step b or do Step c.
   a. **Enter the name of the directory where the command resides into the Pathname field, click the Load button, and do Step b.**
   b. **Expand the name of the directory where the command resides, highlight the name of the command, and move it to the included list.**
   c. **Drag the icon for the command from its folder into the included list.**
4. **To specify security attributes, highlight the name of the command in the included list, and click on the appropriate buttons.**
   Specify privileges in Step 5.
5. **Specify privileges, if desired.**
   a. **Click the Set Privileges button.**
   b. **On the Set Privileges dialog box, move the desired privileges to the included list.**

c. Click OK.

6. Specify a label range, if desired.
   a. Click the Set Minimum SL button.
   b. On the Set Minimum SL dialog box, specify a minimum SL.
   c. Click OK.
   d. Click the Set Maximum SL button.
   e. On the Set Maximum SL dialog box, specify a maximum SL.
   f. Click OK.
7. Specify effective UID/GID, if desired.
   a. Click the Set UID/GID button.
   b. On the Set UID/GID dialog box, move the desired user name to the included list.
   c. Move the desired group name to the included list.
   d. Click OK.
8. On the Profile Manager, choose Save from the Profiles menu.
9. If you are finished, choose Close from the Profiles menu.
10. If you are not finished, choose Actions or Authorizations from the View menu.

### ▼ To Specify Actions in an Execution Profile

1. If the Profile Manager is not up, bring it up as described in "To Access the Profile Manager" on page 223 and "To Pick a Naming Service and Filter for Profiles" on page 224.
2. Choose Actions from the View menu.
3. Put the name of the action you want to specify into the included list.
4. If the action is not in an application group in the default list, drag the icon for the action from its folder into the included list.
   a. Highlight the name of the action in the included list.

5. **Specify security attributes for an action of the appropriate type, if desired.** If the type is TT_MSG, the buttons for specifying security attributes are grayed and you cannot specify security attributes for the action. If the type is COMMAND, click on the appropriate buttons to specify security attributes, as described in Step 6, Step 7, and Step 8.
6. **Specify privileges, if desired.**
   a. **Click the Set Privileges button.**
   b. **On the Set Privileges dialog box, move the desired privileges to the included list.**
   c. **Click OK.**
7. **Specify a label range, if desired.**
   a. **Click the Set Minimum SL button.**
   b. **On the Set Minimum SL dialog box, build or type in a minimum SL.**
   c. **Click OK.**
   d. **Click the Set Maximum SL button.**
   e. **On the Set Maximum SL dialog box, build or type in a maximum SL.**
   f. **Click OK.**
8. **Specify effective UID/GID, if desired.**
   a. **Click the Set UID/GID button.**
   b. **Move the desired user name to the included list.**
   c. **Move the desired group name to the included list.**
   d. **Click OK.**
9. **Choose Save from the Profiles menu.**
10. **If you are finished, choose Close from the Profiles menu.**
11. **If you are not finished, choose Commands or Authorizations from the View menu.**

### ▼ To Specify Authorizations in an Execution Profile

1. **If the Profile Manager is not up, bring it up as described in "To Access the Profile Manager" on page 223 and "To Pick a Naming Service and Filter for Profiles" on page 224.**
2. **Choose Authorizations from the View menu.**
3. **Put the name of the authorization you want to specify into the included list by highlighting its name and moving it to the included list.**
4. **Choose Save from the Profiles menu.**
5. **If you are finished, choose Close from the Profiles menu.**
6. **If you are not finished, choose Commands or Actions from the View menu.**

# *Part 3 — Managing Hosts and Networks*

Most hosts running Trusted Solaris are connected to a network. Installing a non-networked standalone host is covered in the *Trusted Solaris Installation and Configuration* manual. This part of the *Trusted Solaris Administrator's Procedures* manual contains the the needed background and procedures for configuring hosts and networks. sharing files, and managing communications among hosts on local and remote networks.

Chapter 9, "Trusted Solaris Concepts for Managing Hosts and Networks" provides definitions of needed concepts.

Chapter 10, "Specifying Security Attributes in Trusted Network Databases and Setting Up Routing" provides the following main topics:

| |
|---|
| *Goals of Trusted Networking* |
| *Trusted Network Databases* |
| *Setting Up Trusted Routing* |

Chapter 11, "Managing Files and File Systems" includes the following main topics:

| |
|---|
| *Specifying Mount Time Security Attributes* |
| *Trusted Solaris and NFS* |

Chapter 12, “Managing NIS+” includes the following main topics:

| |
|---|
| *Managing Multiple Trusted Solaris Hosts in a Security Domain* |
| *Managing Standalone Trusted Solaris Hosts* |
| *New Trusted Solaris NIS+ Tables and Files Not Administered By NIS+* |
| *Adding Trusted NIS+ Tables* |

Chapter 13, “Changing Configurable Trusted Solaris Kernel Switches” includes the following main topics:

| |
|---|
| *Behaviors Controlled by Configurable Trusted Solaris Kernel Switches* |
| *How Kernel Switches Are Set and Changed* |

Chapter 14, “Managing Printing” includes the following main topics:

| |
|---|
| *How Trusted Solaris Features Address Information Labeling and Access Control for Printers* |
| *Assigning Labels to Print Jobs* |
| *Using a Label Range on Printers to Control Which Jobs Can Print* |
| *Labels, Job Numbers, and Handling Information Printed on Banner and Trailer Pages* |

Chapter 15, “Managing Device Allocation and Setting Device Label Ranges” includes the following main topics:

| |
|---|
| *Managing Device Allocation and Setting Device Label Ranges* |

Chapter 16, “Adding Software” includes the following main topics:

| |
|---|
| *Review of Terms and Concepts* |
| *Security Administrator’s Tasks in Adding Software* |
| *Issues Around the Adding of Privileges to Any Software* |
| *Procedures for Adding Software* |

# *Trusted Solaris Concepts for Managing Hosts and Networks* 9

This chapter provides the necessary concepts and background for administering hosts on a network. Other chapters in this part of the manual, referenced below, describe procedures.

The Trusted Solaris software supports network communications between Trusted Solaris 2.x hosts and the following three types of hosts:

- Other Trusted Solaris 2.x hosts
- Hosts running operating systems that do not recognize security attributes (such as Solaris and other UNIX systems)
- Hosts running other trusted operating systems, including Trusted Solaris 1.x hosts.

Networking is managed by a number of different Trusted Solaris subsystems.

- Trusted NFS is used to manage mounted filesystems.

  Mounts among Trusted Solaris 2.x hosts and mounts between Trusted Solaris 2.x hosts and other hosts that recognize NFS are supported. See Chapter 11, “Managing Files and File Systems” for how to set up mounts.

- NIS+ provides centralized management of configuration files defining hosts, networks and users.

  In the Trusted Solaris system, the NIS+ master can only be used to manage Trusted Solaris hosts’ data. See Chapter 12, “Managing NIS+” for how to manage NIS+.

- Solstice AdminSuite is used to centrally manage users and hosts by maintaining administrative data in NIS+ tables on the NIS+ master server (with the option of updating the corresponding local files on a host by host basis, without relying on NIS+).

  To add new workstations (including diskless workstations) and servers to an already-configured distributed system, see the *Trusted Solaris Installation and Configuration* manual. See the other chapters of this manual for how to use the tools in Solstice AdminSuite to administer users and hosts.

- Trusted networking software supports trusted network communications.

  The security administrator role specifies values in the trusted networking databases to provide the desired degree of either openness or control for communications among hosts in a security domain and between security domains. See “Goals of Trusted Networking” on page 248 in Chapter 10, “Specifying Security Attributes in Trusted Network Databases and Setting Up Routing.” See “Trusted Solaris Networking” for more about the concept of security domains.

## *Trusted Solaris Networking*

The simplest, the easiest-to-protect, and the only network configuration that is submitted for government evaluation is a single distributed system, in which one or more hosts are:

- Running the Trusted Solaris 2.5 operating system
- Administered by the same NIS+ master server, with the same set of security attributes for all machines and users, and
- On the same wire (or configured so as to be virtually on the same wire)

Hosts administered this way may be considered to be within the same *security domain*. Within a single security domain, no routing table is needed, because a packet sent out on the wire is picked up by all other hosts on the same network. When a site decides to include other hosts on the local network that are running other operating systems at varying levels of trust, the security administrator should consider providing some measure of protection for the wire.

Physical protection by the customer against tapping of such a network is assumed, because Trusted Solaris software does not provide encryption.

A single host running the Trusted Solaris operating system by itself is also considered to be a standalone security domain. If the standalone Trusted Solaris host is connected to a network with unlabeled hosts and hosts running trusted software from other vendors, it needs the same types of entries in its trusted networking databases that are needed on any other Trusted Solaris host.

## *Example of a Single Security Domain*

Figure 9-1 shows a single security domain with host F as the NIS+ master. All of the hosts are connected to the network by means of a single network interface. The *network interface* is the physical connector on a host that connects it to a network cable. Most often, the interface is an Ethernet connector to an Ethernet cable that links all the hosts in the local network. The *network accreditation range* for a host or for a network interface (which applies to all communications through the interface) is a *range of sensitivity labels* (SLs) or a *label range*. The network accreditation range is defined by a minimum sensitivity label and a maximum sensitivity label, which are decided upon by the security administrator and specified by the security administrator role in the appropriate trusted networking databases.

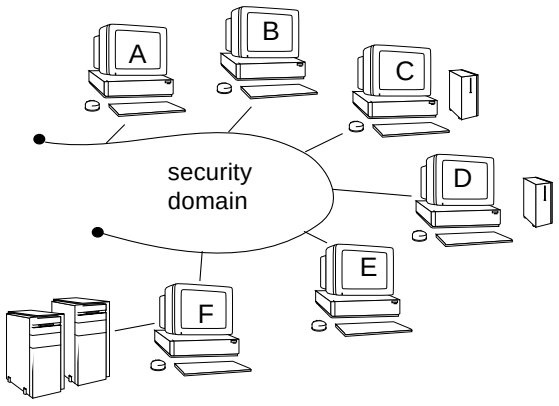



*Figure 9-1* A Single Security Domain

## Example of Multiple Security Domains

Multiple security domains made up of multiple NIS+ domains may be connected by gateways. Figure 9-2 on page 238 shows two security domains. Host F is the NIS+ master for security domain #1, and Host K is the NIS+ master for security domain #2. The two security domains are connected by gateway host G.

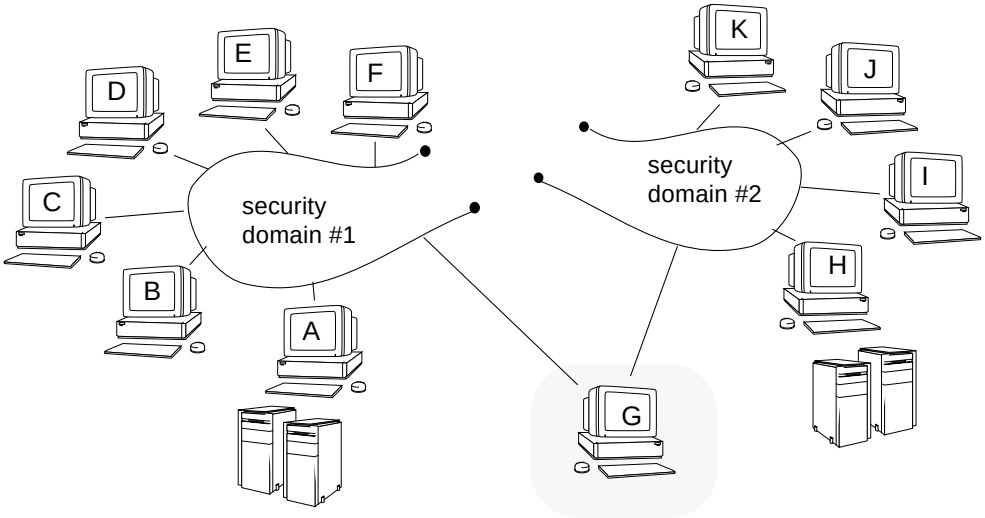



*Figure 9-2* Two Security Domains

In Trusted Solaris administration terms, a *gateway* is usually a host that has more than one network interface and that is running Trusted Solaris software.

## Static Routing and Issues About the Use of Trusted Solaris Hosts As Routers

Dynamic routing is not supported in the Trusted Solaris 2.5 system. As long as all hosts are using static routing, multiple networks may be connected using Trusted Solaris gateways, which is the recommended configuration. Using a Trusted Solaris gateway allows the trusted networking software to use the attributes specified in the trusted network databases to enforce security policy on packets coming into and leaving the network and to support trusted routing of packets that have CIPSO and RIPSO labels.

Communications to a Trusted Solaris distributed system through a Trusted Solaris gateway fail from hosts that rely on dynamic routing because the Trusted Solaris host does not advertise itself as a router to the outside world, so there is no way for1gd hosts using dynamic routing to find the gateway.

To allow communications with host or hosts that rely on dynamic routing (such as Solaris hosts outside of the control of the Trusted Solaris security administrator), a router not running Trusted Solaris can be used. If so, keep in mind that the trusted networking features, which include the trusted routing of CIPSO and RIPSO packets, are not available when the router is other than a Trusted Solaris host.

## *Network Accreditation Range Requirements*

For most hosts running Trusted Solaris software, the accreditation range is specified in the minimum sensitivity label and maximum sensitivity label entries for the host in the `tnrhdb(4TSOL)` file as ADMIN_LOW to ADMIN_HIGH. With this accreditation range, each host on a homogeneous Trusted Solaris distributed system can receive packets from any other host in the same system at any valid sensitivity label. A gateway may be set up either to allow communications at all sensitivity labels or to restrict communications to a subset of all labels with hosts outside of the local distributed system.

The *primary* interface that connects the gateway to the Trusted Solaris network must always have the default network accreditation range of ADMIN_LOW to ADMIN_HIGH, but the security administrator may choose to specify other, restricted, network accreditation ranges for *secondary* interfaces that connect the gateway to other networks. Restricting a network's accreditation range in the trusted networking interface database on a gateway is one of the means by which the security administrator configures the appropriate degree of openness or control for communications at a site.

The security administrator role can restrict communications by specifying a lower maximum sensitivity label or a higher minimum sensitivity label or both in the `tnidb` interface definition for one or more interfaces on a gateway machine.

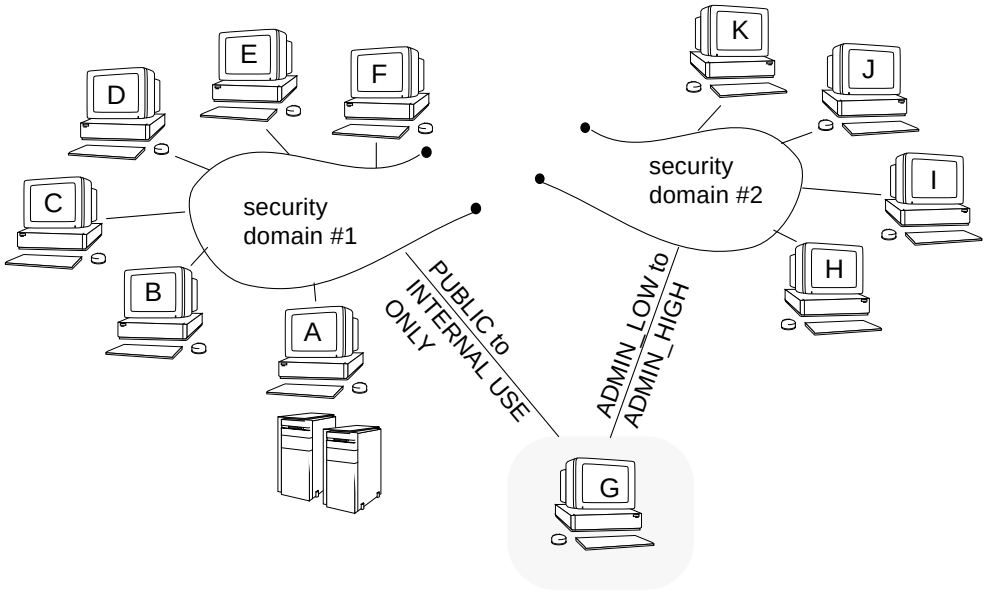



*Figure 9-3* Two Security Domains With Differing Accreditation Ranges

To restrict communications the `tnidb` entry on gateway G in Figure 9-3 on page 240 could be specified as shown in Table 9-1

*Table 9-1* Example tnidb Entry to Restrict a Network's Accreditation Range

| Interface Name | Network Accreditation Range |
|---|---|
| le0 (interface to security domain #1) | PUBLIC to INTERNAL USE ONLY |
| le1 (interface to security domain #2) | ADMIN_LOW to ADMIN_HIGH |

## *Static Routing*

For communications outside of the local area network, the Trusted Solaris administrator must specify gateways in one of two static routing tables in the `/etc` directory on each host. Two types of gateway entries can be created.

- Network gateways

  A network entry routes all communications to the specified network exclusively through the named gateway.

- Default gateways

  Routing table entries for default gateways allow communications through the named gateway to all networks that are not specifically listed.

**Note –** Dynamic routing is not supported.

Both network gateways and default gateways can be specified in the `tsolgateways(4TSOL)` file. A routing table with only network entries would restrict communications only to the specified network(s). A routing table with one or more network entries can also contain a default entry.

Default gateways for simple networks can be specified in the `/etc/defaultrouter` file [which is described in the `route(1MTSOL)` man page]. The trusted networking software looks first for `tsolgateways`, and if one exists, the `defaultrouter` file is not consulted. If neither file exists, no routing is possible.

Any gateways specified in the static routing tables must be directly connected to the local area network to which the Trusted Solaris host is connected. The number entered into the metric field of the `tsolgateways` file is equal to the number of gateways between the originating host and the destination.

A transmission to a host on the same network has 0 hops. Figure 9-4 shows four hosts (H1, H2, H3, and H4) on the same network with 0 hops.

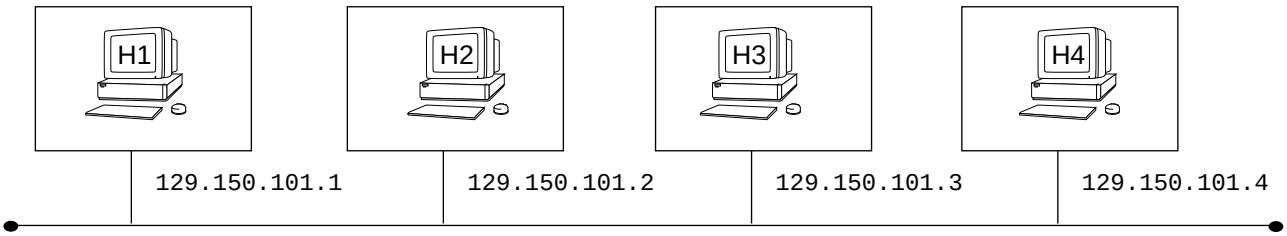



*Figure 9-4* Example of 0 Hops for Communications Between Four Hosts in a Single Security Domain

A transmission to a network that is connected directly to the gateway of the originating network has 1 hop. Figure 9-5 shows network `129.150.101.0` connected directly to `129.150.102.0` by gateway G. Gateway G has a host name for each of its interfaces: gateway-101 is the name for the interface

connected to the 101 network, and gateway-102 is the name for the interface connected to the 102 network. Since `/etc/defaultrouter` can be used for simple networks with only one gateway, hosts 1, 2, 3, and 4 on net `129.150.101.0` could have a `defaultrouter` file specifying only gateway-101. If the `tsolgateways` file is used, the network address needs to be specified as `129.150.102.0` and the metric would be specified as 1 hop.

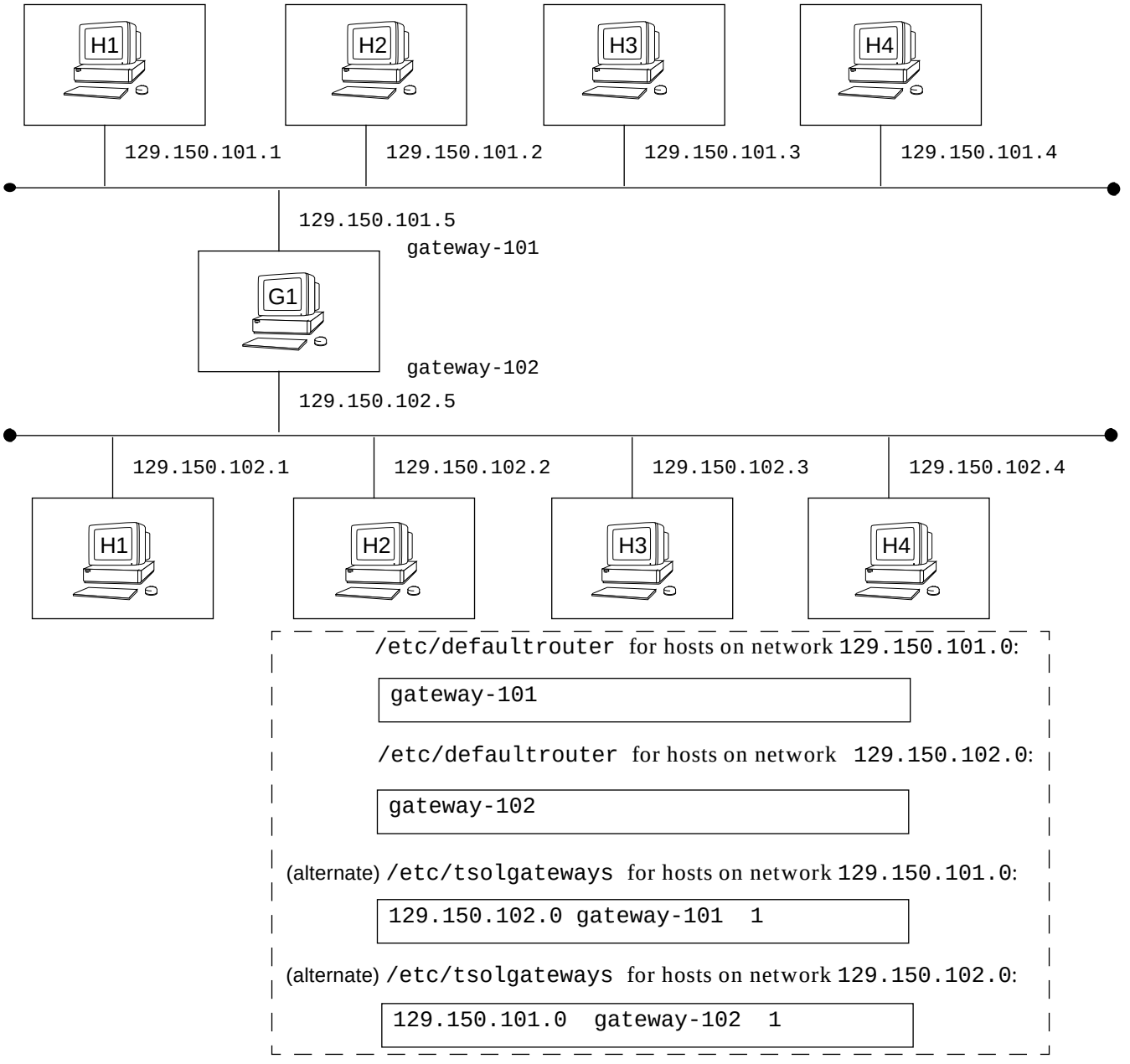



*Figure 9-5* Example: Default and Network Routes for Two Security Domains with a Single Gateway

Transmission to a network two gateways away has two hops as shown in Figure 9-6 on page 245. Figure 9-6 shows an example `tsolgateways` file for hosts 1, 2, 3, and 4 on the network whose IP address is `129.150.110.0`. As illustrated, gateway G1 connects network `129.150.101.0` to network `129.150.102.0`, and gateway G2 connects network `129.150.102.0` to network `129.150.103.0`. The first entry in `tsolgateways` sets a network route to `129.150.102.0` through gateway1-101, and the metric 1 indicates one hop to the destination network. The second entry sets a network route to `129.150.103.0` through gateway1-101, and the metric 2 indicates two hops to the destination network.

Figure 9-7 on page 246 shows a more complex network topology, with gateways G1, G2, and G3.

See “Setting Up Trusted Routing” on page 271 and also “Setting Up Trusted Routing” on page 271 and the associated procedures in “To Set Up a Simple Default Route for a Network with One Gateway” on page 297 and “To Set Up Trusted Routing” on page 297, all of which are in Chapter 10, “Specifying Security Attributes in Trusted Network Databases and Setting Up Routing.”

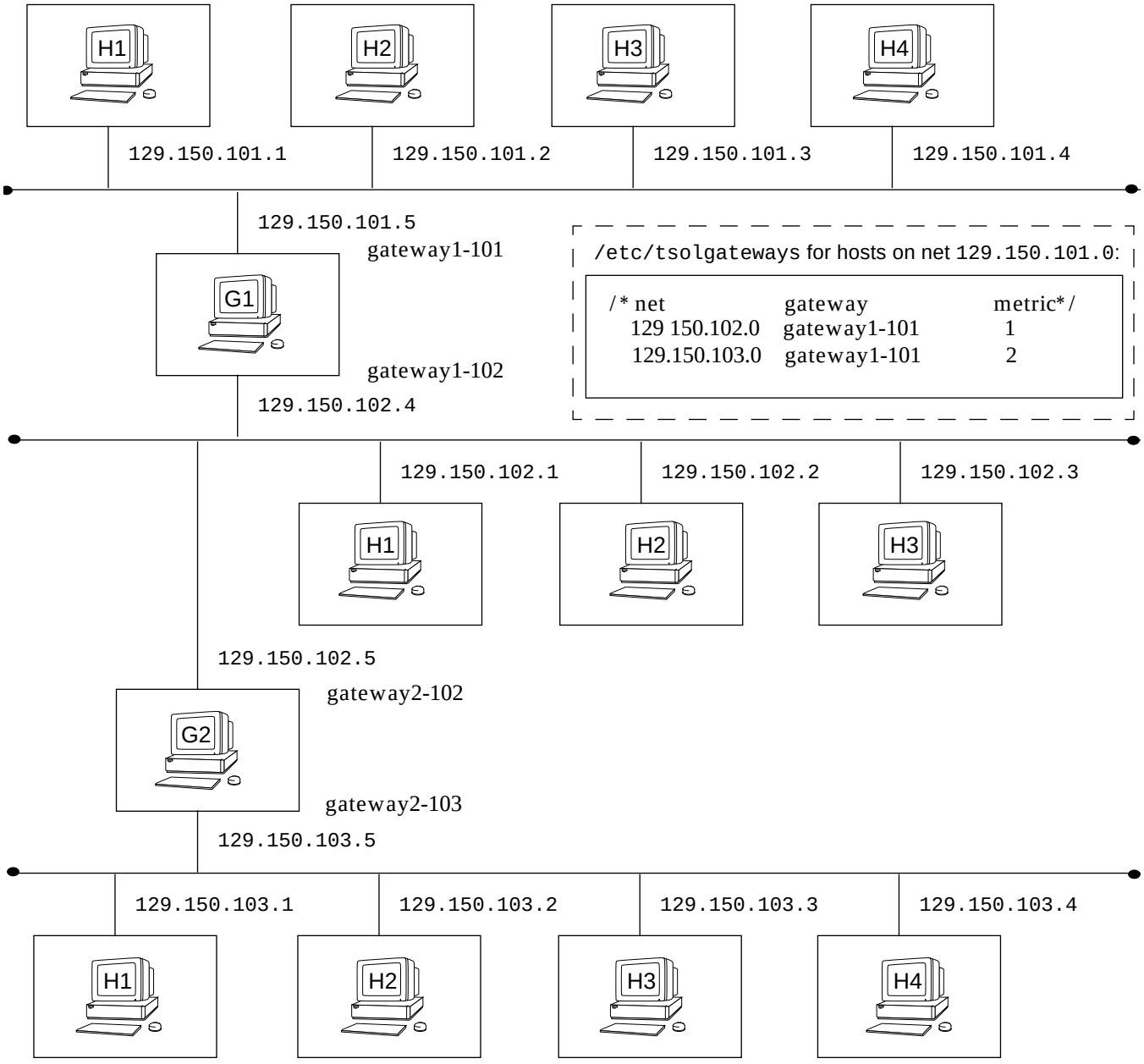



*Figure 9-6* Example tsolgateways File for Communications Among Three Networks

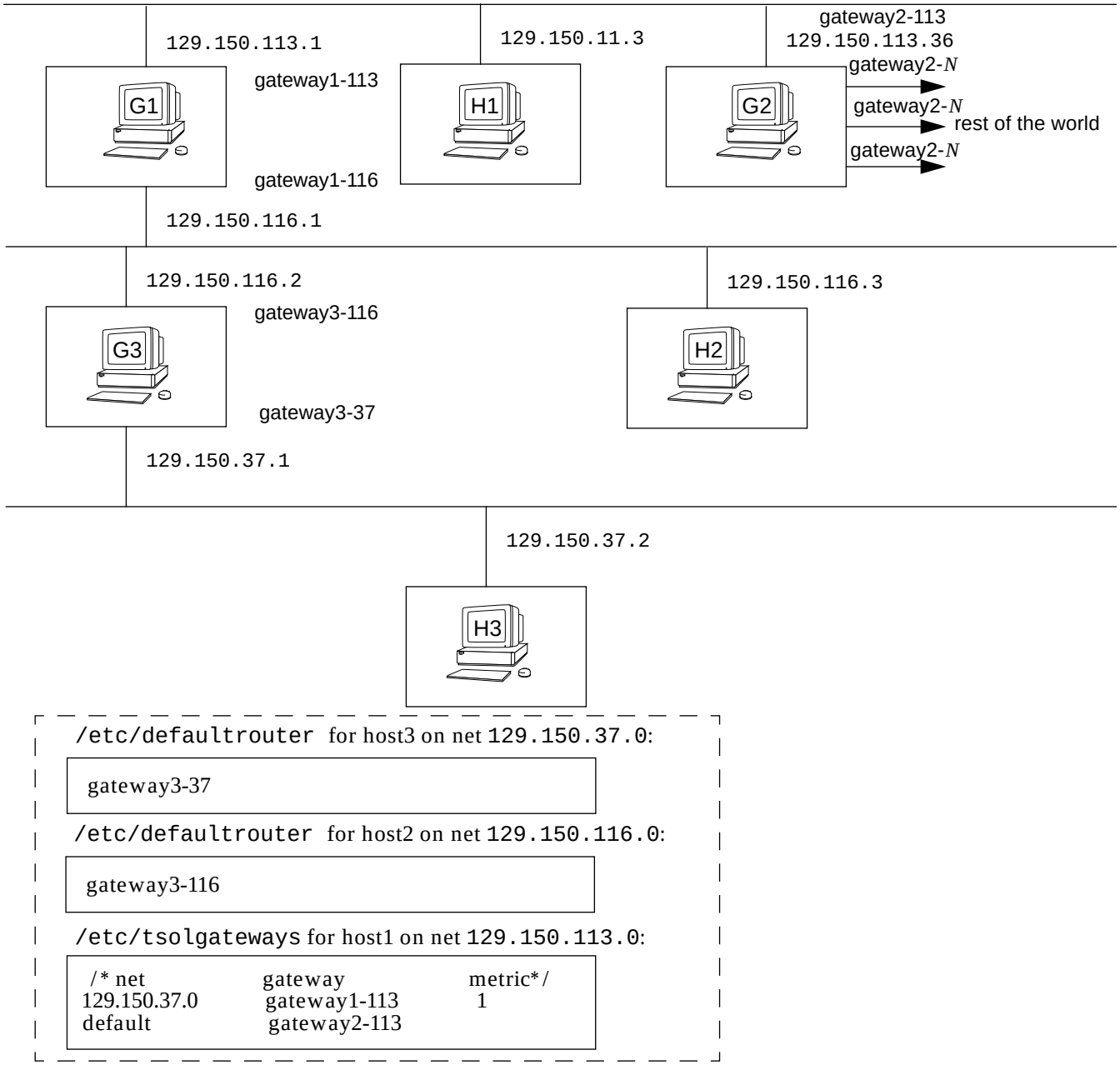



*Figure 9-7* Example Complex Gateway Configuration With Routing Tables

# *Specifying Security Attributes in Trusted Network Databases and Setting Up Routing*

*10*

The Trusted Solaris distributed system can consist either of a single host or a number of hosts networked together. The security administrator specifies security attributes in trusted network databases that define the Trusted Solaris system's view of all hosts and networks with which communications are allowed. The security administrator also sets up routing.

This chapter provides procedures for the security administrator to follow when specifying which hosts are allowed to communicate with the Trusted Solaris system and which security attributes apply to communications with these hosts. This chapter provides the following information:

- The background needed for understanding what values should be set for various security attributes in the trusted networking databases
- How the values set in the trusted networking databases are put to use
- How to set up routing and trusted routing

These listed sections provide needed prerequisite background information:


| | |
|---|---|
| *Goals of Trusted Networking* | *page 248* |
| *How Security Attributes Are Added to Packets* | *page 249* |
| *MAC Enforcement on Outgoing Messages* | *page 252* |
| *MAC Enforcement on Incoming Messages* | *page 253* |
| *Trusted Network Databases* | *page 254* |

# 10


| | |
|---|---|
| *Host Types and Trusted Networking* | *page 255* |
| *Setting Up Trusted Routing* | *page 271* |
| *Creating Entries in the Trusted Networking Databases* | *page 273* |


This chapter includes the following procedures for updating the various databases to achieve the desired degree of isolation or openness in communications.


| | |
|---|---|
| *To Access the Trusted Network Databases from the Database Manager* | *page 280* |
| *To Assign a Template to a Single Host in the tnrhdb* | *page 282* |
| *To Assign a Template to a Group of Hosts in the tnrhdb* | *page 284* |
| *To Create a Wildcard tnrhdb Entry for All Hosts Not Otherwise Specified* | *page 287* |
| *To Set an Accreditation Range in a Host Template or Network Interface Entry* | *page 290* |
| *To Configure a Network Interface* | *page 291* |
| *To Add a New Entry or Modify an Existing Entry in tnidb(4TSOL)* | *page 292* |
| *To Set Up a Simple Default Route for a Network with One Gateway* | *page 297* |
| *To Set Up Trusted Routing* | *page 297* |


## *Goals of Trusted Networking*

The trusted networking software ensures the following:

- Proper labeling of data in network communications
- Enforcement of MAC rules when data is sent or received across the network

The `tnrhdb(4TSOL)`, `tnrhtp(4TSOL)`, and the `tnidb(4TSOL)` trusted network databases store labels and other security attributes that apply to hosts and networks and to network interfaces.

The trusted network databases also may be used by the security administrator to specify that packets from a Trusted Solaris machine are routed to certain specified hosts only through gateways that have a security level that is consistent with the sensitivity of the data being transmitted. This type of routing is called:

- Trusted routing

## How Security Attributes Are Added to Packets

The topic of how security attributes are added to packets is a prerequisite for understanding the definitions of host types and for understanding how to specify security attributes in the trusted network databases. In general, network packets have the format shown in Figure 10-1:

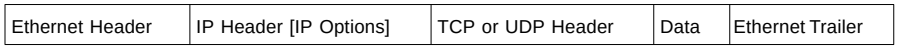



*Figure 10-1* Packet Format

Both TCP/IP and UDP/IP can be used for network communications. When a host is specified with the Trusted Solaris 2.5 host type or the TSIX host type, the trusted networking software inserts a SAMP header after the TCP or UDP Header and before the data. Therefore, when the host type is Trusted Solaris 2.5 or TSIX, the packets have the format shown in Figure 10-2:

*Figure 10-2* TSIX and Trusted Solaris 2.5 Packet Format

The options field of the IP header (shown as [IP Options] in Figure 10-2) is used for putting RIPSO or CIPSO labels and options in packets.

### IP Options

The IP options field is used to carry the two supported types of IP labels: RIPSO and CIPSO labels. These labels may be specified by the Trusted Solaris administrator to set up trusted routing. (See “Setting Up Trusted Routing” on page 271.)

Incoming packets carry RIPSO or CIPSO labels and options when the sending host has a RIPSO or CIPSO host type.

Outgoing packets carry IP labels and other options in their IP options field:

- If the entry for a host with either the Trusted Solaris 2.x or TSIX host type has its IP Label Type field set to RIPSO or CIPSO
- If the entry for a host is identified with a host type of either CIPSO or RIPSO.

## CIPSO Labels in Packets

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

- Assigns the host the CIPSO Host Type
- Assigns the host the Trusted Solaris 2x 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 applied to outgoing packets is derived by the trusted networking software from the actual Trusted Solaris sensitivity label that is associated with the data. For example, at a site running the same labels defined in the default `label_encodings` file, a packet whose sensitivity label is C would be given a CIPSO label of CONFIDENTIAL. A direct mapping of the sensitivity label of CONFIDENTIAL to the CIPSO label of CONFIDENTIAL is possible in this case, but most Trusted Solaris sensitivity labels probably do not map directly to CIPSO labels. For this reason, the security administrator should plan the site's labels to map well to CIPSO labels if the site plans to use CIPSO labels for trusted routing or wishes to communicate with a host with a host type of CIPSO.

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

- The sending host
- All gateway hosts through which messages travel and
- The destination host

As explained in *Trusted Solaris Label Administration*, the classification portion of the ADMIN_HIGH sensitivity label is 32767 bits and the compartments portion is 256 bits, all turned on. By default, a message is dropped if it is sent at a sensitivity label that is too big to map to a CIPSO label. The Trusted Solaris 2.x ADMIN_HIGH sensitivity label is one example of a sensitivity label too big to map to a CIPSO label.

The security administrator can add a switch to the `/etc/system` file to choose what action should be taken when a packet has the ADMIN_HIGH sensitivity label, whether:

- The label should be mapped to a CIPSO label with the highest classification and all compartments turned on, or
- The packet should be dropped.

See “To Substitute a Valid CIPSO Label for the ADMIN_HIGH Sensitivity Label” on page 296.

## *RIPSO Labels in Packets*

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 Trusted Solaris 2x Host Type, setting the IP Label Type to RIPSO
- Assigns the host the TSIX Host Type, setting the IP Label Type to RIPSO

RIPSO labels on outgoing packets are administratively defined. The security administrator 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.

Table 10-1 shows the supported RIPSO Send classifications.

*Table 10-1* Supported Classifications for RIPSO Labels

| **Supported Classifications for RIPSO Labels** |
|---|
| Top_Secret |
| Secret |
| Confidential |
| Unclassified |

The RIPSO Send PAF and Return PAF fields refer to Protection Authority Flags, which are shown in Table 10-2. 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 by receiving hosts in response to incoming RIPSO labeled packets. The classification sent back in the ICMP message is the same as the RIPSO classification in the packet.

*Table 10-2* Protection Authority Flags That May Be Specified in the RIPSO Send PAF or as RIPSO Return PAF Fields

| **Protection Authority Flags (may be used with RIPSO labels or specified as RIPSO errors)** |
|---|
| GENSER |
| SIOP-ESI |
| SCI |
| NSA |
| DOE |

## *MAC Enforcement on Outgoing Messages*

Accreditation range checks by the network software involve checks of the specified sensitivity label range and checks of IP labels. (See “How Security Attributes Are Added to Packets” on page 249, “CIPSO Labels in Packets” on page 250, and “RIPSO Labels in Packets” on page 251 for more about IP labels.)

The sensitivity labels of outgoing messages are compared to the accreditation range of all of the following:

- The destination host [as specified in the template assigned to the host in `tnrhdb(4TSOL)`/`tnrhtp(4TSOL)`]
- All gateways (if any) through which the message must pass [as specified in the gateways’s entry in`tnrhdb(4TSOL)`/`tnrhtp(4TSOL)`]

  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.

If the check is being done on a gateway and the message is being forwarded to another gateway, another first hop check is performed based on the network accreditation range of the gateway to which the message is being forwarded.

- The network interface (as specified in the local `tnidb(4TSOL)`)

  If the sensitivity label of a message is not within the minimum and maximum sensitivity 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*

For incoming messages, the Trusted Solaris networking software obtains sensitivity 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 system. In many cases, messages arrive from hosts that are not label-cognizant or that do not send recognizable labels, or the messages do not have all of the other required attributes in their packets. In some other cases, the security attributes are not in accessible portions of the packet (see “How Security Attributes Are Added to Packets” on page 249).

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. The labels and other security attributes in the trusted network databases are specified by each site’s security administrator role and are defined according to the security administrator’s assessment of the level of trust of the sending host. Any attributes not obtainable from the host’s entry are supplemented by the attributes specified in the entry in the trusted network interface database that applies to the `interface` through which the message arrives.

The sensitivity label of the incoming packet is obtained from:

1. **The default sensitivity label supplied in the template assigned to the sending host.**
2. **The accreditation range assigned in the `tnidb` file to the interface through which the message comes.**

If the packet has a CIPSO label in the IP Header, the CIPSO label for the sending host is compared to the CIPSO label of the gateways and of the destination hosts.

## *Trusted Network Databases*

The security administrator uses the Trusted Solaris version of the Solstice Database Manager, shown in Figure 10-3, to make entries in the `tnrhdb(4TSOL)`, `tnrhtp(4TSOL)`, and `tnidb(4TSOL)` trusted network databases.

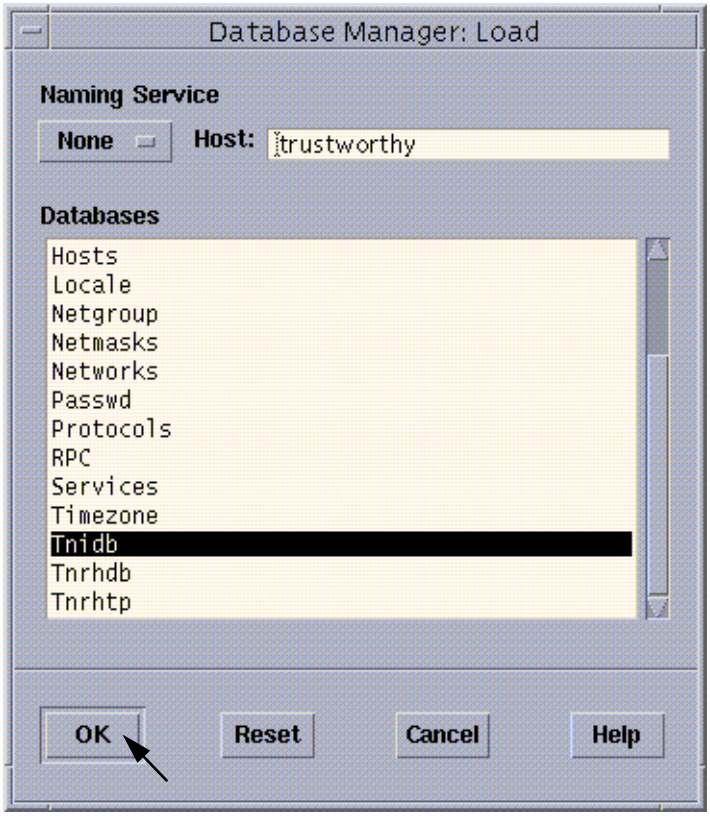



*Figure 10-3* Tnidb Selected in the Database Manager: Load List

- The `tnrhtp` file stores templates that define sets of security attributes that apply to various hosts types
- The `tnrhdb` file is used to assign templates from the `tnrhtp` file to hosts and networks
- The `tnidb` file is used to assign security attributes to network interfaces.

The values specified in the `tnidb` file are consulted along with any attributes specified in the applied to the `tnrhtp/tnrhdb` files for individual hosts and networks to determine which security attributes apply to communications that come into the distributed system through the interface. See “MAC Enforcement on Incoming Messages” on page 253 and “Precedence Rules for Attributes in Trusted Network Databases” on page 276.

Before any of the trusted networking files is edited using the Database Manager, a naming service must be selected. The two choices are:

- NIS+
- None

Because the `tnidb` file’s values apply to the network interfaces on the local host, the `tnidb` file is *always* a local file in `/etc/security/tsol`.

The `tnrhdb` and `tnrhtp` files are usually administered as NIS+ maps. However, a standalone Trusted Solaris host may be configured with no naming service, and on such a host with no naming service, these files are kept in `/etc/security/tsol`.

The kernel caches all the information from `tn*db` databases. When no naming service is selected, after any changes are made, the kernel cache is automatically updated on the local host. When the NIS+ naming service is selected, it can take up to 1/2 hour for any `tn*db` changes to be pulled by NIS+ clients from the NIS+ master.

## *Host Types and Trusted Networking*

In the entries made in a combination of the `tnrhdb` and `tnrhtp` databases, multiple hosts and networks may be associated with a specific *host type*. The host type identifies the *protocols* used by the kernel for interpreting a message, and the protocols tell the kernel which security attributes to look for in the packet header.

# 10

## Host Types Supported

Table 10-3 lists the host types that can be specified in a template.

*Table 10-3* Supported Host Types

| **Tnrhtp Template Manager Field** | **Name in `tnrhtp` Database** |
|---|---|
| `Trusted Solaris 2.x` | `sun_tsol` |
| `TSIX` | `tsix` |
| `MSIX` | `msix` |
| `CIPSO` | `cipso` |
| `RIPSO` | `ripso` |
| `Unlabeled` | `unlabeled` |

**Note –** Trusted Solaris hosts may be specified as any of the supported host types.

**Note –** Once a host type has been selected in the Template Manager, fields that are not settable for the host type are grayed out and not active.

The sections listed below describe the supported host types with figures that show the security attributes that are supported for each type.


| | |
|---|---|
| *Trusted Solaris 2.x (sun_tsol) Host Type* | *page 257* |
| *TSIX (tsix) Host Type* | *page 260* |
| *MSIX (msix) Host Type* | *page 263* |
| *CIPSO (cipso) Host Type* | *page 265* |
| *RIPSO (ripso) Host Type* | *page 267* |
| *Unlabeled (unlabeled) Host Type* | *page 269* |

### *Trusted Solaris 2.x (*sun_tsol*) Host Type*

| Ethernet Header | IP Header [IP Options] | TCP or UDP Header | SAMP Header [Binary Attributes] | Data | Ethernet Trailer |
|---|---|---|---|---|---|

When the sun_tsol host type is specified, the trusted networking software uses the SAMP protocol for sending security attributes. The security attributes are represented in binary form, without token mapping.

The set of supported security attributes for the sun_tsol host type is not as large as the set supported by the tsix host type or the set supported on TS 1.x systems: missing are supplementary groups and audit information. (In TS 1.x, audit information includes: def_audit ID, terminal ID, and audit preselection mask; other trusted systems provide other audit information.)

**Note –** Token mapping is used for communications when hosts are specified with the TSIX host type, which makes it possible to configure Trusted Solaris hosts as TSIX hosts if you wish them to communicate with a full set of security attributes.

**Note –** Diskless clients of Trusted Solaris servers must always be specified with the sun_tsol host type.

If the IP Label Type field is set in a Trusted Solaris 2.x type host's entry to CIPSO or RIPSO, IP options in the IP Header are used to carry CIPSO or RIPSO labels that can be used for trusted routing of communications from Trusted Solaris 2.x hosts. See "Setting Up Trusted Routing" on page 271. Also see "IP Options" on page 249.

## *Security Attributes Specifiable for the Trusted Solaris 2.5 Host Type*

The security attributes that can be specified for Trusted Solaris 2.5 hosts are shown in Figure 10-4 on page 258 and explained in Table 10-4 on page 259. Fields that are not settable for the host type are grayed out and not active.

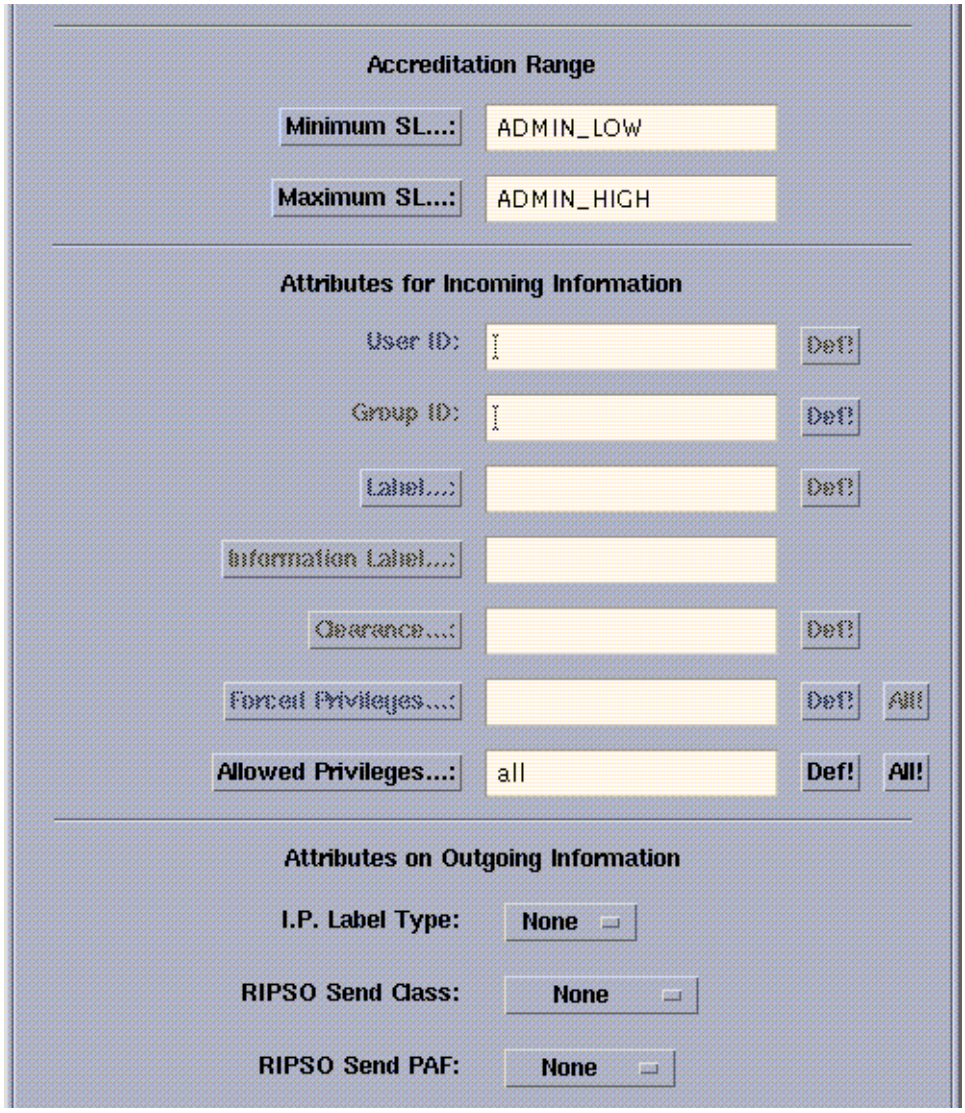



*Figure 10-4* Configurable Fields for the sun_tsol Host Type in the Tnrhtp

*Table 10-4* Required and Optional Security Attributes for sun_tsol Host Types

| Configurable Attributes | Notes |
|---|---|
| minimum SL (min_sl=) | Specify ADMIN_LOW or use another valid sensitivity label to restrict communications with the sun_tsol host. |
| maximum SL (max_sl=) | Specify ADMIN_HIGH or use another valid sensitivity label to restrict communications with the sun_tsol host. |
| allowed privileges (allowed_privs=) | Specify to limit the effective privilege set propagated across the network by this host (both inbound and outbound). Only specified privileges are allowed. “all” means that all privileges are interpreted; “none” means that no privileges are interpreted. In the tnrhtp template that applies to the host, an empty field (allowed_privs=;) means that whatever is specified in the `tnidb` for this field applies. In the `tnidb` file, an empty field means that no privileges are applied. |
| IP label (ip_label=) | Optional. Valid types are: none, ripso, cipso |
| RIPSO label (ripso_label=) | Optional. Must be specified if IP Label Type is RIPSO. Supported Classification Level Encodings: Top_Secret, Secret, Confidential, Unclassified. Supported Protection Authority Flags: GENSER, SIOP-ESI, SCI, NSA, DOE. |
| RIPSO error (ripso_error=) | Optional. Must be specified if IP Label Type is RIPSO. Supported Protection Authority Flags: GENSER, SIOP-ESI, SCI, NSA, DOE. The Classification Level is taken from the ripso_label field. |
| CIPSO DOI (cipso_doi=) | Optional. Must be specified if IP Label Type is CIPSO. A number corresponding to the host’s Domain of Interpretation for CIPSO labeled packets. |

## *TSIX (`tsix`) Host Type*

When a TSIX host type is specified, the trusted networking software communications with that host observe the TSIX standard established by the Trusted Systems Interoperability Group (TSIG), of which Sun Microsystems Federal, Inc. is a member. For TSIX host types, the trusted networking software uses the SATMP token mapping protocol. An Attribute header in the SAMP header specifies that the attributes are in token form. Except that attributes are transmitted in token form rather than in binary form, the packets have the same format as the sun_tsol packets.

In a TSIX host's entry, the IP Label Type field can be set to none or CIPSO or RIPSO. IP options in the IP Header are then used to carry CIPSO or RIPSO labels to be used for trusted routing. See "Setting Up Trusted Routing" on page 271.

To communicate using the TSIX host type, both the sending and receiving hosts must be within the same tsix DOT (domain of translation).

TSIX-type hosts that have the IP Label Field specified as CIPSO *must* have the `tsol_admin_high_to_cipso` switch in the `/etc/system` file set to 1 or the TSIX host cannot communicate with Trusted Solaris hosts. See Chapter 13, "Changing Configurable Trusted Solaris Kernel Switches" for more details.

## *Security Attributes Specifiable for TSIX Hosts*

**Note –** Sun's definition of the TSIX token maps is the first definition of a TSIX DOT, and other companies who wish to communicate with Trusted Solaris systems need to use the Sun-defined DOT—at least until more than one TSIX DOI is defined. There may be a TSIX DOI registry in the future to define what the tokens mean.

The trusted networking software assigns any missing security attributes on incoming packets from TSIX hosts by getting defaults from the trusted networking databases.

The security attributes that can be specified for TSIX hosts are shown in Figure 10-5 on page 261 and explained in Table 10-5 on page 262. Fields that are not settable for the host type are grayed out and not active.

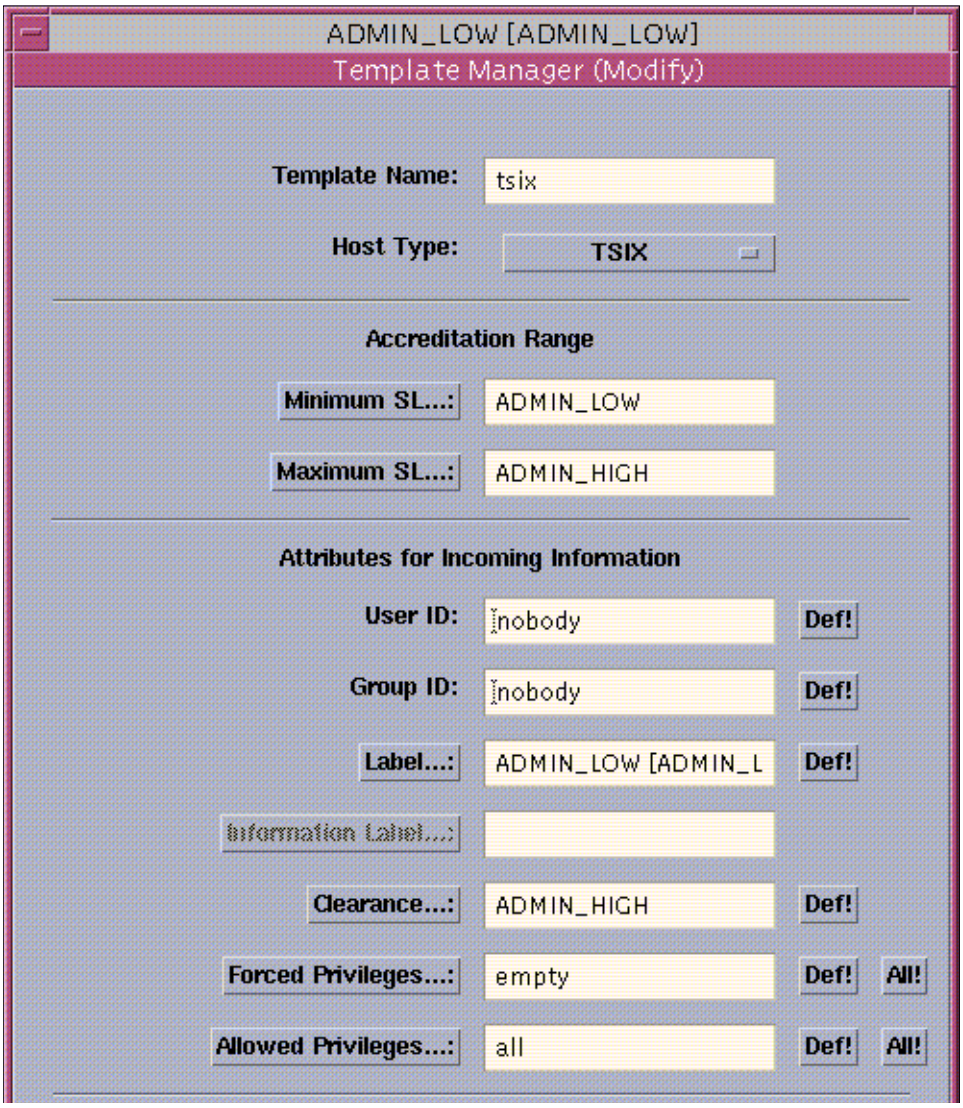



*Figure 10-5* Configurable Fields for the tsix Host Type in the Tnrhtp

*Table 10-5* Required and Optional Security Attributes for tsix Host Types *(1 of 2)*

| Configurable Attributes | Notes |
|---|---|
| minimum SL (min_sl=) | Specify in hexadecimal format ADMIN_LOW or use another valid sensitivity label to restrict communications with the tsix host. |
| maximum SL (max_sl=) | Specify in hexadecimal format ADMIN_HIGH or use another valid sensitivity label to restrict communications with the tsix host. |
| allowed privileges (allowed_privs=) | Optional. Specifies limits for the effective privilege set propagated across the network (both inbound and outbound) by this host. Only specified privileges are allowed. “all” means that all privileges are interpreted; “none” means that no privileges are interpreted. In the `tnrhtp` template that applies to the host, an empty field (`allowed_privs=;`) means that whatever is specified in the `tnidb` for this field applies. In the `tnidb` file, an empty field means that no privileges are applied. |
| forced privileges (forced_privs=) | Optional. The defined privileges are applied to incoming packets received from the host being defined, because it does not supply privileges. “all” means that all privileges are applied; “none” means that no privileges are applied. In the tnrhtp template that applies to the host, an empty field (`forced_privs=;`) means that whatever is specified in the `tnidb` for this field applies. In the `tnidb` file, an empty field means that no privileges are applied. |
| default label (def_label=) | Optional. The specified label is applied to incoming packets received from the host being defined, because it does not supply privileges. If no information label is specified, an information label of ADMIN_LOW is applied. |
| default clearance | Optional. The specified label is applied to incoming packets received from the host being defined, because it does not supply a clearance. |
| default UID | Optional. The specified UID is applied to incoming packets received from the host being defined, because it does not supply a UID. |

*Table 10-5* Required and Optional Security Attributes for tsix Host Types *(2 of 2)*

| Configurable Attributes | Notes |
|---|---|
| IP label (ip_label=) | Optional. Valid types are: NONE, RIPSO, CIPSO |
| RIPSO label (ripso_label=) | Optional. Required if IP Label Type is RIPSO. Supported Classification Level Encodings: Top_Secret, Secret, Confidential, Unclassified. Supported Protection Authority Flags: GENSER, SIOP-ESI, SCI, NSA, DOE. |
| RIPSO error (ripso_error=) | Optional. Required if IP Label Type is RIPSO. Supported Protection Authority Flags: GENSER, SIOP-ESI, SCI, NSA, DOE. The Classification Level is taken from the ripso_label field. |
| CIPSO DOI (cipso_doi=) | Optional. Required if IP Label Type is CIPSO. A number corresponding to the host's Domain of Interpretation for CIPSO labeled packets. |

## *MSIX (*`msix`*) Host Type*

| Ethernet Header | IP Header [IP Options] | TCP or UDP Header | Data | Ethernet Trailer |
|---|---|---|---|---|

Labels are the only security attributes supported in communications from MSIX hosts, and they are carried in the IP header.

The MSIX host type should be specified in templates for hosts running TS 1.x, which includes MSIX 1.0 plus some features of MSIX 2.0. Because Trusted Solaris MSIX does not support all of the features of MSIX 2.0, it should not be specified in templates for hosts that use the MSIX 2.0 protocol.

The security attributes that can be specified for TSIX hosts are shown in Figure 10-6 on page 264. Fields that are not settable for the host type are grayed out and not active.

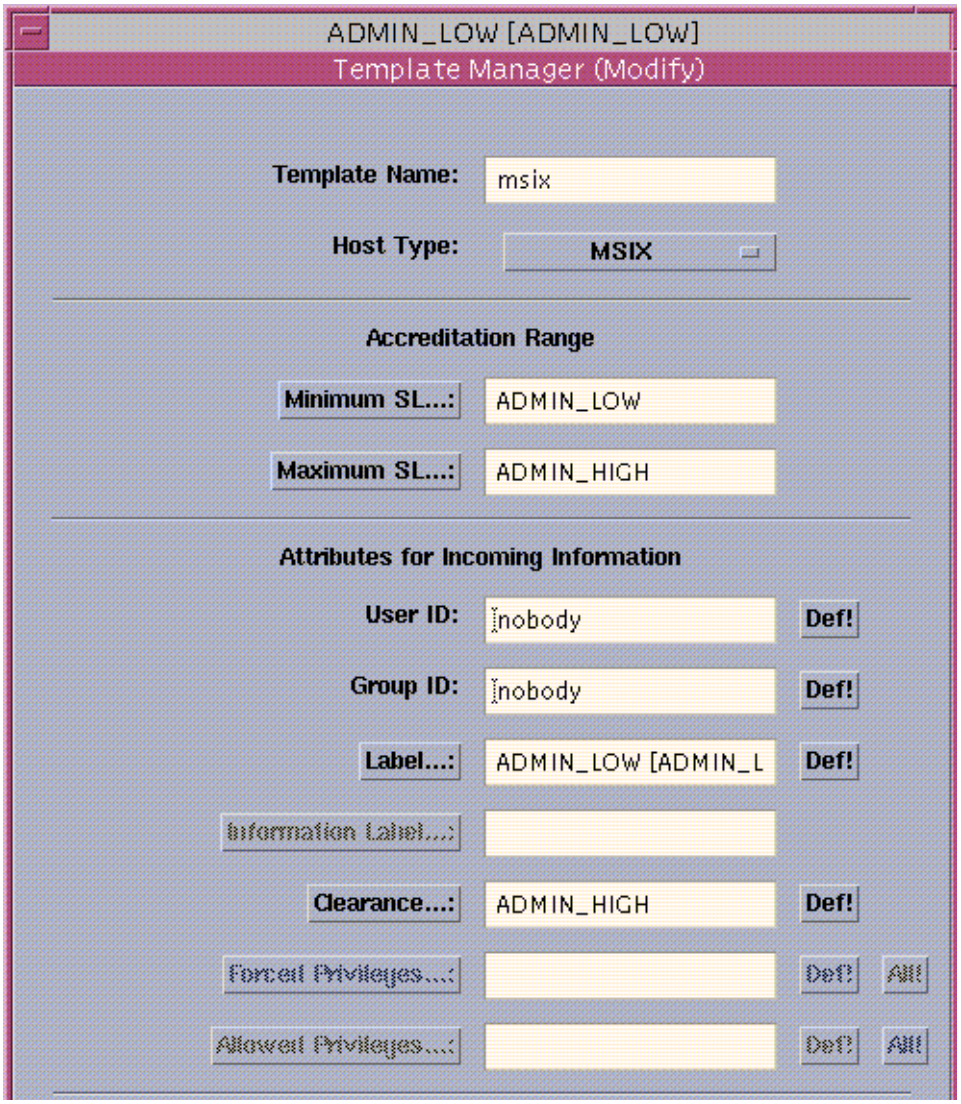



*Figure 10-6* Configurable Fields for the msix Host Type in the Tnrhtp

## *CIPSO (*`cipso`*) Host Type*

| Ethernet Header | IP Header [CIPSO Label in IP Options] | TCP or UDP Header | Data | Ethernet Trailer |
|---|---|---|---|---|

The CIPSO host type supports communications with hosts that conform to the CIPSO standard. The only security attribute supported for CIPSO hosts is a CIPSO label, which is derived from the sensitivity label of the data and inserted into the IP header of packets going to hosts of host type CIPSO. The host entries must specify the same domain of interpretation (CIPSO DOI) as that of the destination host and of all gateways through which the packets from that host are routed. If the sending host is specified as a CIPSO host type, the trusted networking software looks for a CIPSO label in the IP Header of incoming packets.

The security attributes that can be specified for TSIX hosts are shown in Figure 10-7 on page 266. Fields that are not settable for the host type are grayed out and not active.

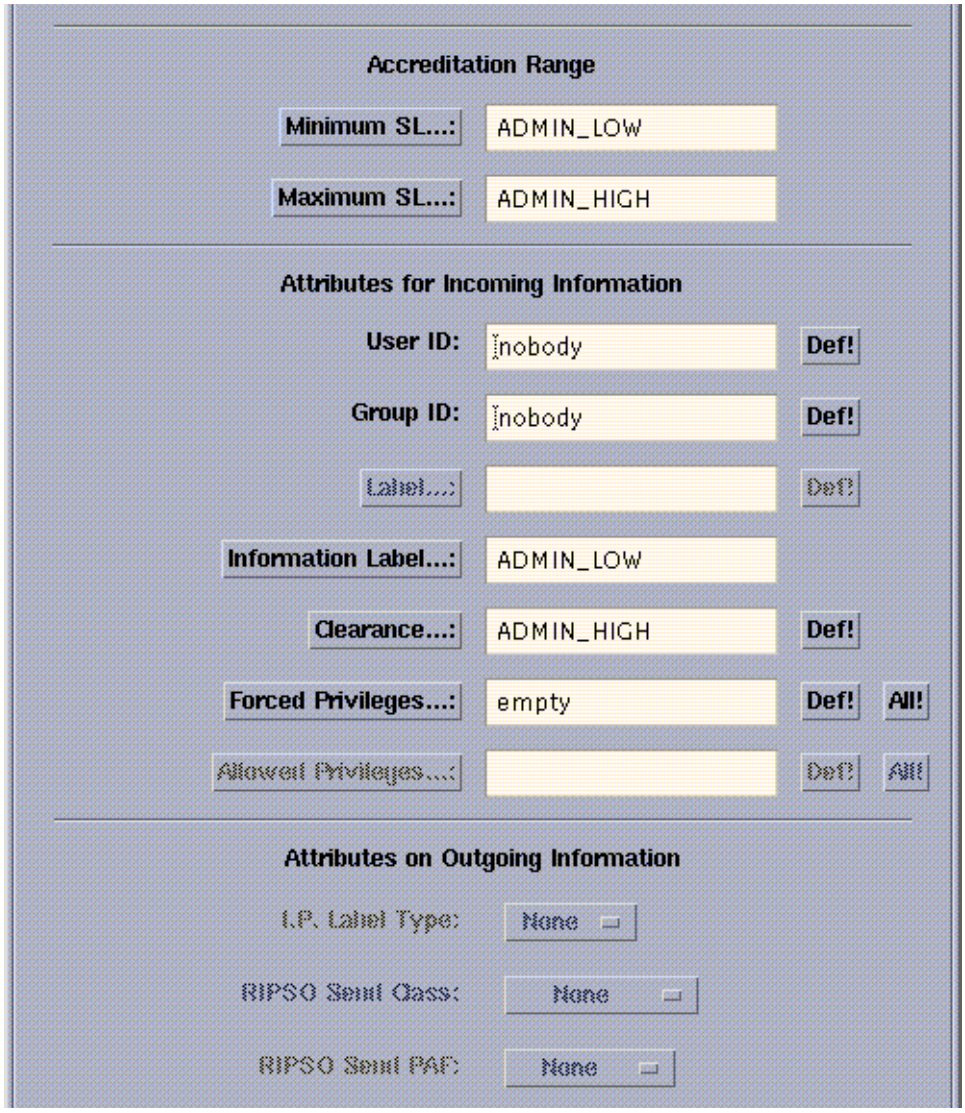



*Figure 10-7* Configurable Fields for the cipso Host Type in the Tnrhtp

## *RIPSO (`ripso`) Host Type*

| Ethernet Header | IP Header [RIPSO Label in IP Options] | TCP or UDP Header | Data | Ethernet Trailer |
|---|---|---|---|---|

Supports hosts that conform to the RIPSO standard. The RIPSO labels that are used for communications with a host of type RIPSO are administratively defined in the template for the host, as specified in the `tnrhtp(4TSOL)` man page. Templates for hosts of type RIPSO must have the RIPSO Send Class set to one of Top Secret, Secret, Confidential, Unclassified, or a hexadecimal valu. Templates must also have both the RIPSO Send PAF and Return PAF set to one of GENSER, SIOP-ESI, CI, NSA, DOE or a hexadecimal value.

The security attributes that can be specified for RIPSO hosts are shown in Figure 10-8 on page 268. Fields that are not settable for the host type are grayed out and not active.

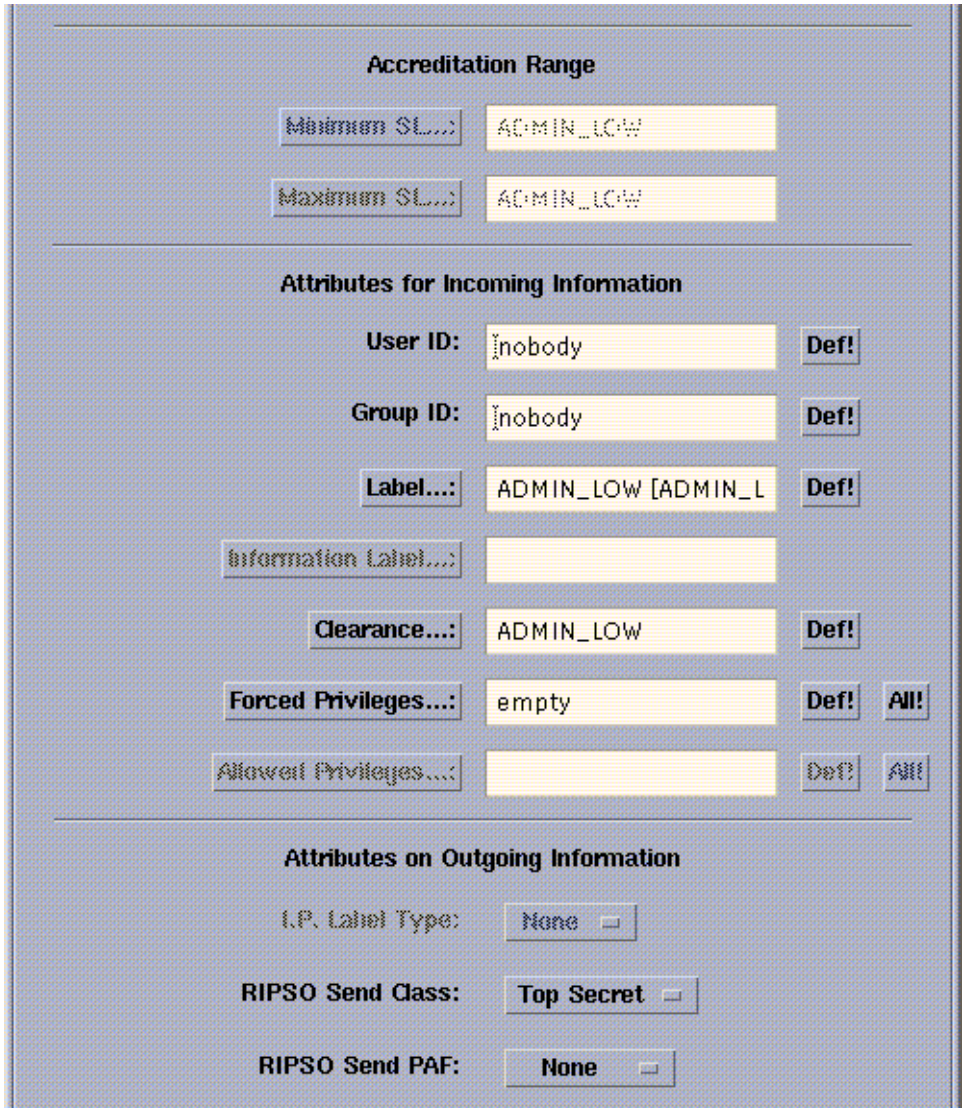



*Figure 10-8* Configurable Fields for the RIPSO Host Type in the Tnrhtp

## *Unlabeled (*`unlabeled`*) Host Type*

| Ethernet Header | IP Header [RIPSO Label in IP Options] | TCP or UDP Header | Data | Ethernet Trailer |
|---|---|---|---|---|

The security attributes that can be specified for unlabeled hosts are shown in Figure 10-9 on page 270. Fields that are not settable for the host type are grayed out and not active.

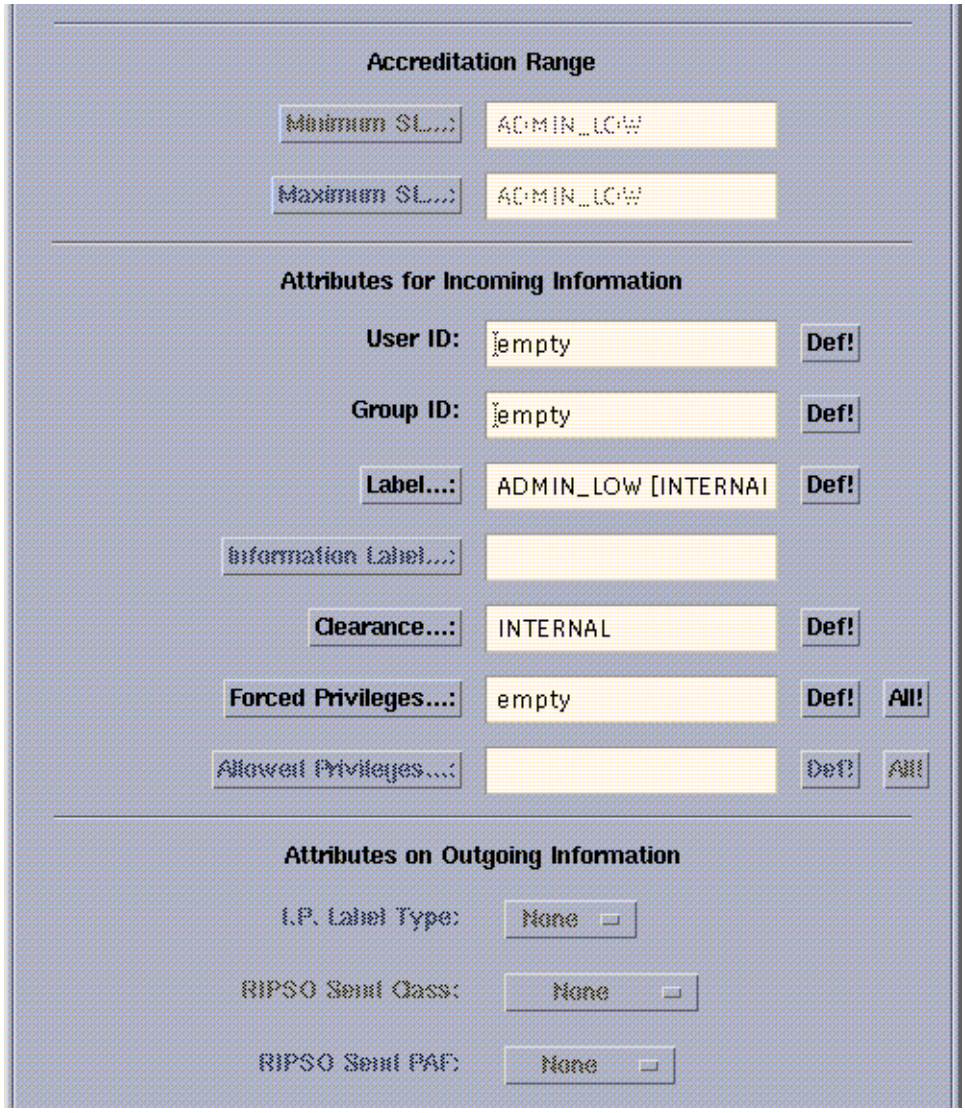



*Figure 10-9* Configurable Fields for the unlabeled Host Type in the Tnrhtp

## Setting Up Trusted Routing

Trusted Solaris allows the system administrator to configure one of two general types of gateways in static routing tables:

- Per network gateways
- Default gateways

As described in Chapter 9, “Trusted Solaris Concepts for Managing Hosts and Networks,” the trusted networking software looks for any network gateways and default gateways that a site may specify in the `tsolgateways(4TSOL)` file before looking for default gateways specified in the `/etc/defaultrouter` file [which is described in the `route(1MTSOL)` man page]. If neither file exists, no routing is done.

This section describes what you need to know to ensure routing of outgoing communications so that they only go through gateways that are configured with a security level that matches the sensitivity of the data being sent out. This section assumes you have read and understood “Static Routing” on page 240 of Chapter 9.” See “To Set Up Trusted Routing” on page 297 of this chapter for the procedure.

The security administrator role begins by specifying entries with the same host type and with the same IP label type for:

- the sending host,
- all gateways, and
- the destination host

The host type must be either:

- Trusted Solaris 2.x or
- TSIX and

The IP Label Type must be specified the same, either:

- RIPSO or
- CIPSO.

The CIPSO label is derived by the trusted networking software from the sensitivity label of the packet, while the RIPSO label must be specified by the security administrator in the RIPSO Class and RIPSO Send PAF fields in the template. The CIPSO or RIPSO label of the packet is then consulted by the trusted networking software and used in combination with information from

static routing tables to ensure that message is sent only through gateways at the appropriate level of trust. This combination of Trusted Solaris and TSIX host type with CIPSO or RIPSO labels enables packets to be sent with Trusted Solaris or TSIX security attributes to hosts that understand them while providing either a CIPSO or RIPSO label in a portion of the packet that is accessible the trusted networking software so that it can be used to control the routes the packet takes along the way. The trusted networking software only looks at the portion of the packet shown in Figure 10-10, and the Trusted Solaris and TSIX labels and other security attributes are stored between the TCP / UDP Header and the Ethernet Trailer, while the CIPSO and RIPSO labels are stored in the IP Header portion of the packet:

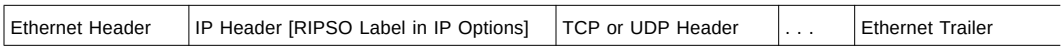



*Figure 10-10* Portions of a Packet Accessible to the Trusted Networking Software

Without the IP label, the trusted networking software on a gateway would not be able to ascertain what a packet's label is and be able to do the necessary accreditation checks on the next hop, because the trusted networking software does not look in the part of the packet where the label and other security attributes are stored. The security administrator role needs to create entries for each of the following specifying them with either the Trusted Solaris 2.x or TSIX host type, and specifying whether to use a CIPSO or RIPSO label (see "To Set Up Trusted Routing"):

- The sending host
- Each of the trusted gateways through which the message should be routed
- The destination host.

**Note –** Because the CIPSO label is obtained from the actual sensitivity label of the data being sent out, the security administrator does not specify a CIPSO label. When the IP label type being used is CIPSO, the security administrators in each of the security domains through which communications using CIPSO labels are being routed must agree among themselves about which Trusted Solaris sensitivity labels are going to be used for communications between them, since the sensitivity label is going to be converted into a equivalent CIPSO label by the trusted networking software. The security administrators also need to agree on the same CIPSO DOI to specify for all the host types in the previous list.

**Note –** Because the RIPSO label and RIPSO error are administratively defined, each of the hosts in the list should have the same RIPSO label and RIPSO error. When the IP label type being used is RIPSO, the security administrators in each of the security domains through which communications using IP labels are being routed must agree among themselves about which RIPSO label, protection authority flags, and RIPSO error to specify for all the host types in the previous list.

The security administrator must also put an entry for each gateway in the `/etc/defaultrouter` file or the `/etc/tsolgateways` file, identifying each gateway either by its IP address or host name. If the hostname is used, the host must be listed in the local `/etc/inet/hosts` file. The router looks at the label in the IP header, looks for in the routing tables for the list of gateways, and then looks in the trusted networking database entries for a gateway from the route table that has a label with the same protocol and the same label as the outgoing packet.

## *Creating Entries in the Trusted Networking Databases*

Before the security administrator can set up the entries in the `tnrhdb`, in which templates are assigned to hosts and networks to templates, he or she should do the following:

- Review the existing templates from the `tnrhtp(4TSOL)` file

  Use the Database Manager to bring up the Tnrhtp database, select each template in turn and view its contents in Template Manager modify dialog box.

- Modify existing templates or create any new templates needed for the site
- Decide which templates should be used for each host and network

### *Using* `tnrhdb` *Options to Achieve a Closed or Open Type of Network Configuration*

Trusted networking supports both the following types of network configurations:

- An *open* configuration permitting *communication with any host*

- A *closely-controlled* configuration permitting communications only with specified hosts and networks

Each site chooses whether to be inclusive or exclusive in allowing communications with other hosts and networks.

### *Hierarchical Fallback Mechanism*

A hierarchical fallback algorithm is used by the trusted networking software in looking for a host's entry in the `tnrhdb(4TSOL)` database, as described in the man page. The trusted networking software first looks for an entry specific to the host, and if it does not find one, it falls back to searching for a matching class C network entry, then a class B entry, a class A entry, and finally a wildcard entry (IP address 0.0.0.0), if one exists. If no entry in the `tnrhdb` database matches the IP address of the host, communications are not allowed with the host. Both open and controlled configurations can make selective use of the fallback mechanism.

### *Open Configuration Using a Wildcard*

The security administrator can make a *wildcard entry* to specify a default set of attributes that will be assigned to any host for which an entry does not exist in the trusted network databases, to allow communications with every host or network. The template assigned to the wildcard entry is configured by the security administrator in the `tnrhtp(4TSOL)` database, along with the other templates, as described under "Setting Up Templates.

### *Closely-controlled Configuration*

Communication with the host is not permitted if the trusted network software cannot resolve a host's IP address to an entry in the `tnrhdb` database. To strictly restrict communications, the security administrator can simply not use a widlcard and be very careful about using network entries.

## Setting Up Templates

See the following procedures, which show some of the different ways the security administrator may associate a host with a template.

| |
|---|
| *To Assign a Template to a Single Host in the tnrhdb* |
| *To Assign a Template to a Group of Hosts in the tnrhdb* |
| *To Create a Wildcard tnrhdb Entry for All Hosts Not Otherwise Specified* |

**Note –** The examples assume that the *tsol_1*, *tsol*, and *wildcard* templates are specified in the `tnrhtp` file.

## Precedence Rules for Attributes in Trusted Network Databases

If the security administrator supplies any value in the `tnrhtp` template for a host, the value in the `tnrhtp` file takes precedence over the values specified for the network interface in the `tnidb`, as shown in Figure 10-11. As shown in Figure 10-11, the values in the `tnidb` apply only when values have not been set in the `tnrhdb` entry for the host sending the packets.

Required fields must be specified in either one or both files. An empty default field in the tnrhtp entry means that the value *must* be supplied in the corresponding default field in the `tnidb` file.

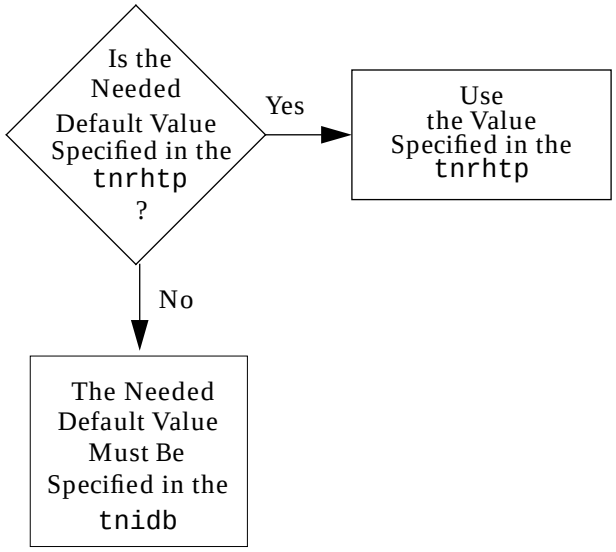



*Figure 10-11* Attribute Precedence Rules

### Example of Combined Entries

In Figure 10-12, a template named wildcard is defined for unknown hosts.

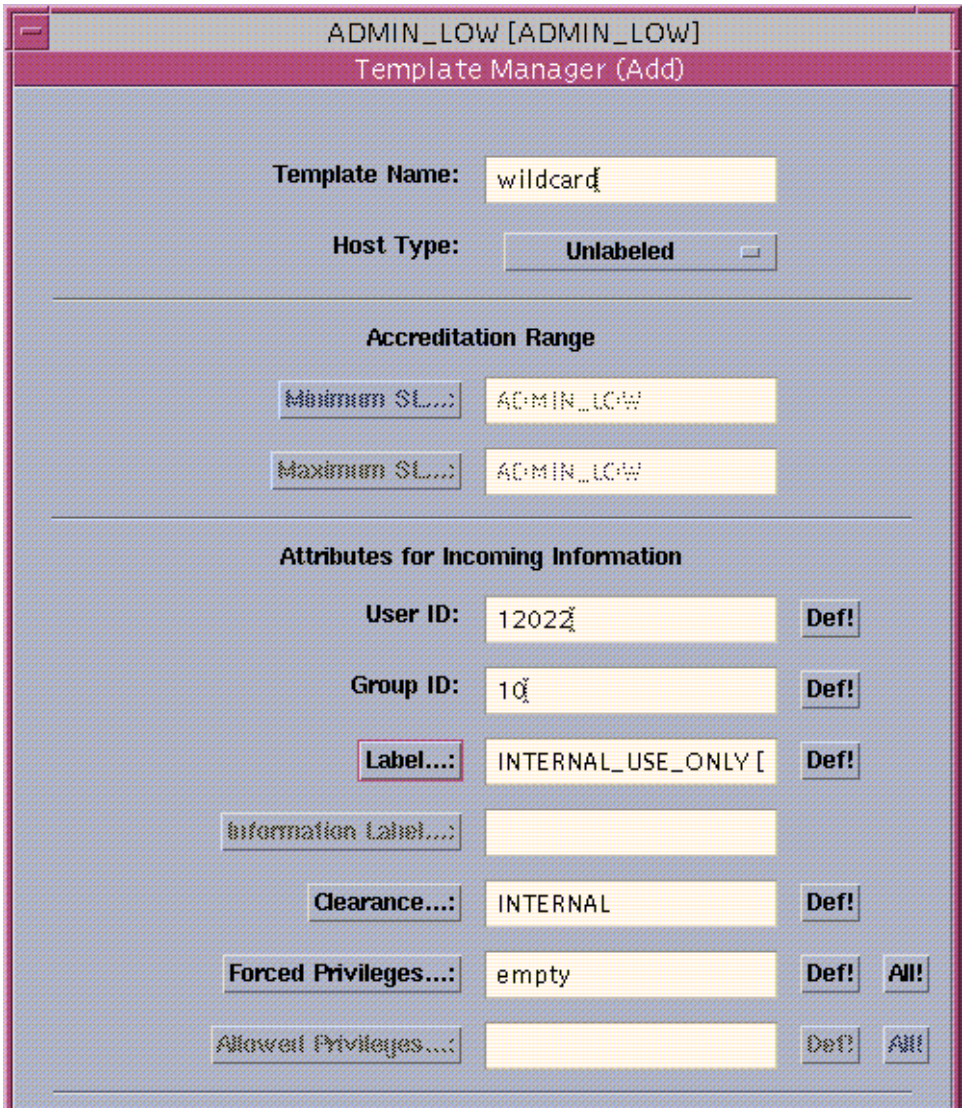



*Figure 10-12* Assigning Some Default Attributes to Communications from Unspecified Hosts

For our example, the default sensitivity label in Figure 10-12 is defined in as INTERNAL_USE_ONLY[INTERNAL], the clearance is INTERNAL, the def_uid is 12022 and def_gid field is 10. Figure 10-13 on page 278 shows a

`tnidb` entry for le0, and let us assume that a packet comes into this network interface from an unknown host, and the attributes for unknown hosts are the ones that are partly defined in the `tnrhtp` default entry in Figure 10-12 on page 277.

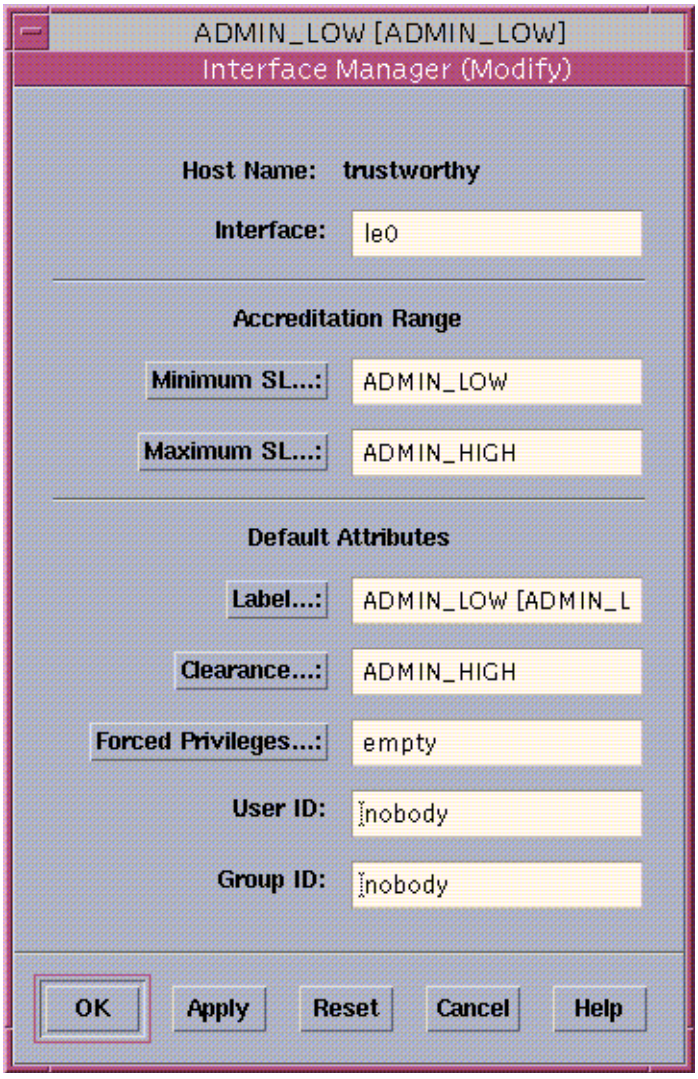



*Figure 10-13* Default Entry for the le0 Interface in the Tnidb Data

The combined effect of the two example entries in Figure 10-12 on page 277 and in Figure 10-13 on page 278 is that any packets coming from an unknown host through the le0 interface are assigned a default label of INTERNAL_USE_ONLY[INTERNAL], a clearance of INTERNAL, a default UID of 120022, and a default GID of 10.

## *Network Accreditation Range*

Each host and network has an accreditation range. The accreditation range is set by a min_sl and a max_sl entry in the `tnrhtp` file for a host and in the `tnidb` file for a network interface.

The example in Figure 10-14 shows host G with two network interfaces, le0 and le1. le0 is defined with an accreditation range of ADMIN_LOW to ADMIN_HIGH, while le1 is defined with an accreditation range of INTERNAL_USE_ONLY to NEED_TO_KNOW ENG.

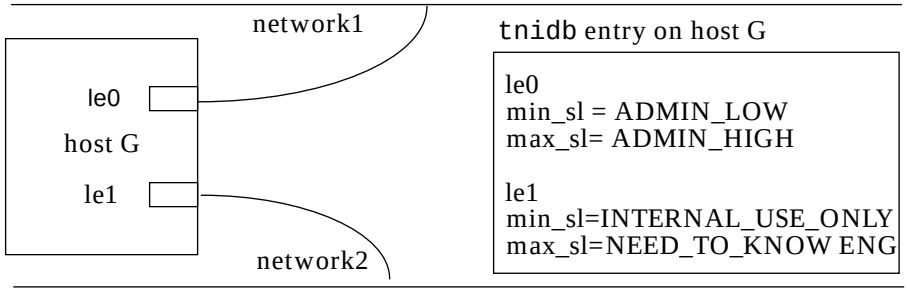



*Figure 10-14* Two Network Interfaces and Their Network Accreditation Ranges

**Note –** Communications within the Trusted Solaris distributed system require an accreditation range of ADMIN_LOW to ADMIN_HIGH.

See “To Configure a Network Interface” on page 291.”

# 10

## *Procedures*

### ▼ To Access the Trusted Network Databases from the Database Manager

1. **Assume the security administrator role.**
2. **In an ADMIN_LOW workspace, open the Solstice Apps folder in the Application Manager.**

Figure 10-15 shows the Database Manager selected in the Solstice_Apps Folder.

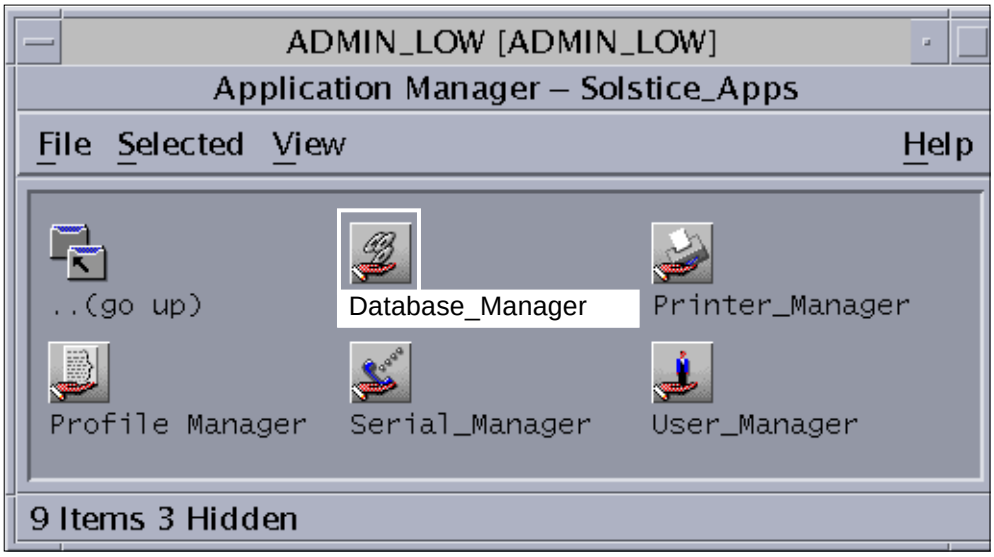



*Figure 10-15* Database Manager Selected in the Solstice_Apps Folder

3. **Double Click on the Database Manager icon to bring up the Database Manager Load List.**
4. **Select a naming service from the Naming Service menu on the Database Manager Load list.**

---

**Note –** By default, as shown in Figure 10-16, the NIS+ naming service is selected, and the name of the NIS+ domain displays in the Domain text entry field.

---

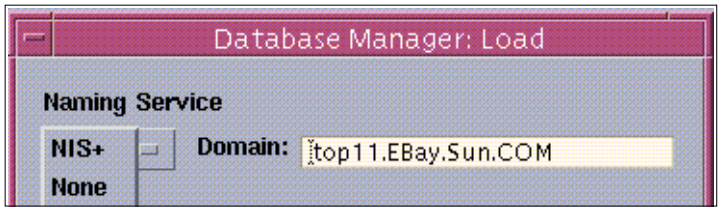



*Figure 10-16* Loading a Naming Service on the Database Manager

---

**Note –** Remember that `tnidb` is always a local file, while `tnrhdb` and `tnrhtp` are managed by NIS+ unless you are configuring a standalone Trusted Solaris system configured with no naming service. If so, all three files are local files that must be modified with Naming Service set to None.

---

**a. For the `tnidb`, choose None.**
Choosing None displays the name of the current host in the Host text entry field, as shown in Figure 10-17.

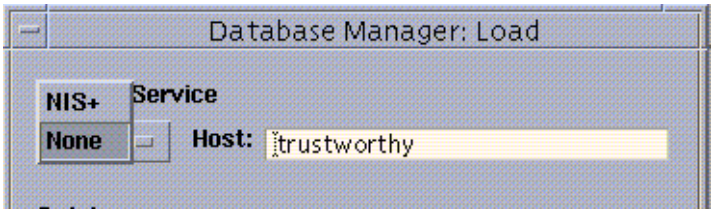



*Figure 10-17* Database Manager Selected in the Solstice_Apps Folder

**b. For the `tnrhdb` and `tnrhtp` files, choose whichever naming service applies, either NIS+ or None.**

5. **If the naming service is None, to configure a database on a remote host, enter the name of a host in the Host text entry field.**

6. **Double click the name of the trusted networking database to be edited or highlight the name and click OK.**
See Figure 10-18 on page 282.

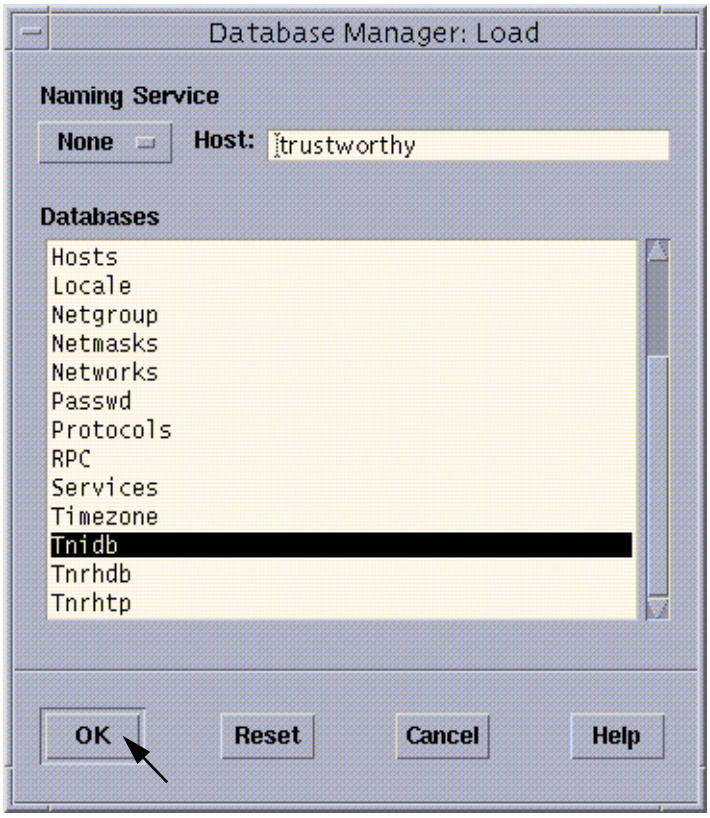



*Figure 10-18* Tnidb Selected in the Database Manager: Load List

## ▼ To Assign a Template to a Single Host in the `tnrhdb`

**Note –** Several fields in the Template Manager dialog box have a default button (marked Def!) to their right. If the security administrator clicks the default button, a default value is supplied in the field.

1. **Access the Database Manager.**
   See "To Access the Trusted Network Databases from the Database Manager" on page 280, if needed.
2. **Decide which of the templates you want to assign out of the `Tnrhtp` database.**

3. **Highlight Tnrhdb from the Database Manager Load list and choose Add from the Edit menu.**

4. **Enter the IP address of the single host and its Template name in the dialog box that displays.**
   Figure 10-19 shows the host identified by its IP address (192.110.120.6) and assigned to the template tsol_1.

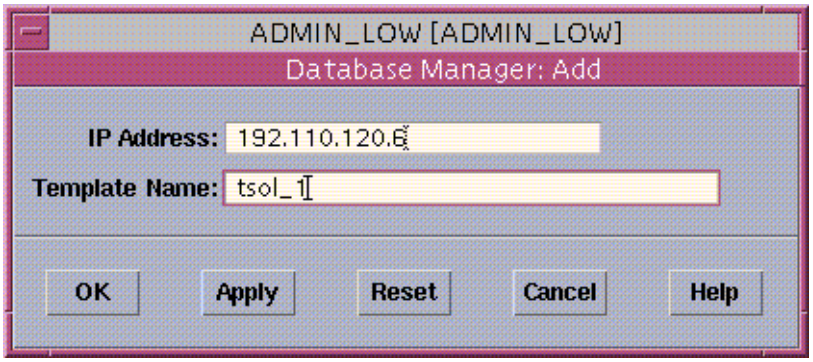



*Figure 10-19* Adding a Host Entry to Tnrhdb and Specifying a Template

5. **Click the OK button at the bottom of the Database Manager: Add menu.**

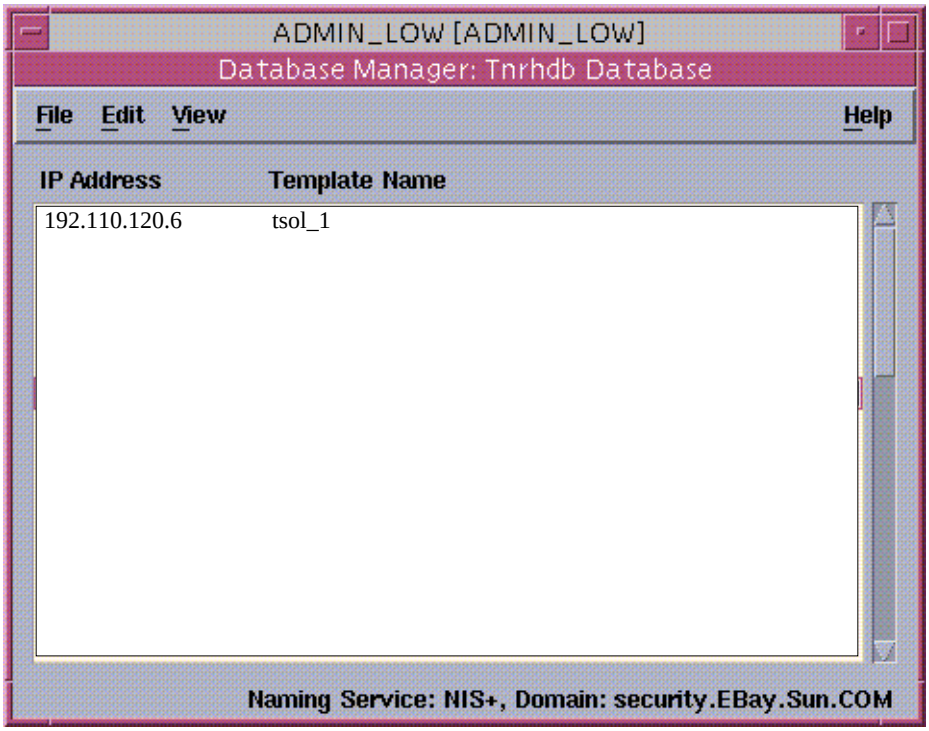



*Figure 10-20* Tnrhdb Host Entry Assigned to the Template Named tsol_1

### ▼ To Assign a Template to a Group of Hosts in the `tnrhdb`

1. **Access the Database Manager.**
   See "To Access the Trusted Network Databases from the Database Manager" on page 280, if needed.
2. **Decide which of the templates you want to assign out of the `Tnrhtp` database.**
3. **Highlight `Tnrhdb` from the Database Manager Load list and choose Add from the Edit menu.**
4. **Enter the IP address of the network and the Template name in the dialog box that displays.**
   Figure 10-19 on page 283 shows a network identified by its IP address (192.110.120.0) and assigned to the template tsol.

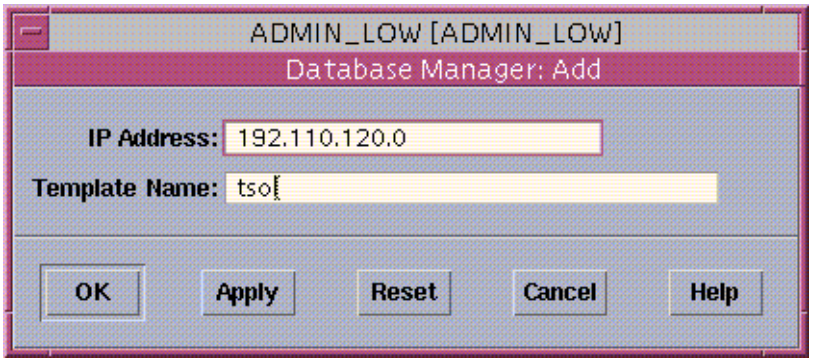



*Figure 10-21* Adding a Network Entry to Tnrhdb and Specifying a Template

Figure 10-22 on page 286 shows a new entry for the network 192.110.120.0, which has the tsol template and an entry for the host whose IP address in 192.110.120.6, which has the tsol_1 template. The host whose IP address in 192.110.120.6 has the tsol_1 template while all others in that network have the tsol template.

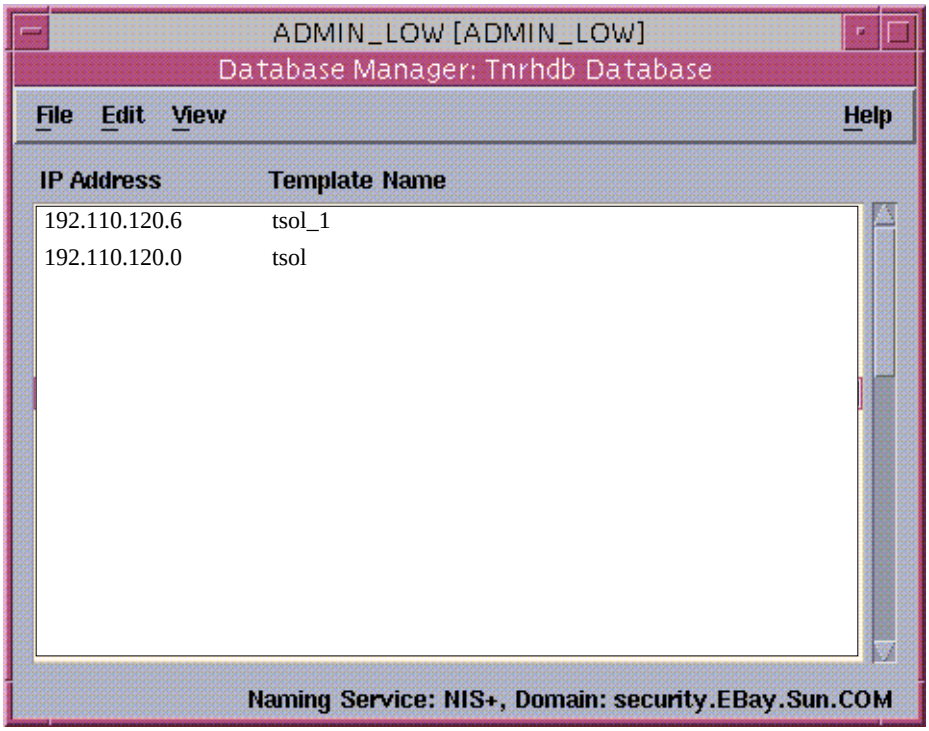



*Figure 10-22* Tnrhdb Network Entry Assigned to the Template Named tsol

---

**Note –** With the two lines shown in Figure 10-22, only hosts on the 192.110.120.0 network can communicate with the security domain (when tnrhdb is a NIS+ table) or with the individual host (when `tnrhdb` is a local file).

---

**Note –** Listing every host—either explicitly or implictly (by making an entry for the network to which the host is connected)—creates a controlled configuration where only the listed hosts are allowed to communicate with the system.

---

**5. Click the OK button at the bottom of the Database Manager: Add menu.**

### ▼ To Create a Wildcard `tnrhdb` Entry for All Hosts Not Otherwise Specified

1. **Access the Database Manager.**
   See "To Access the Trusted Network Databases from the Database Manager" on page 280, if needed.

2. **Decide which of the templates you want to assign out of the `Tnrhtp` database.**

3. **Highlight `Tnrhdb` from the Database Manager Load list, and choose Add from the Edit menu.**
   See "To Access the Trusted Network Databases from the Database Manager" on page 280, if needed.

4. **Enter the wildcard IP address of 0.0.0.0 and the Template name in the dialog box that displays.**
   Figure 10-23 assigns the wildcard template to the `Tnrhdb` wildcard entry.

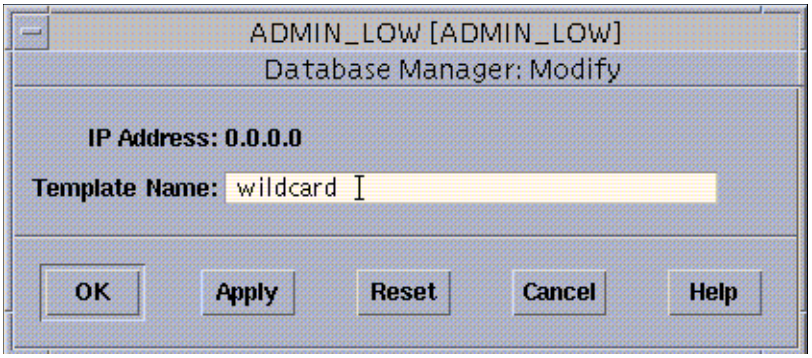



*Figure 10-23* IP Address and Template Name for a Tnrhdb Fallback Template Entry

Figure 10-24 on page 288 shows a new entry for the wildcard IP address, associating it with a template called wildcard. The new wildcard entry assigns all hosts not already-specified to the default entry.

---

**Note –** Using a wildcard entry creates a wide open configuration where *any host* is allowed to communicate with the system.

---

5. **Click the OK button at the bottom of the Database Manager: Add menu.**

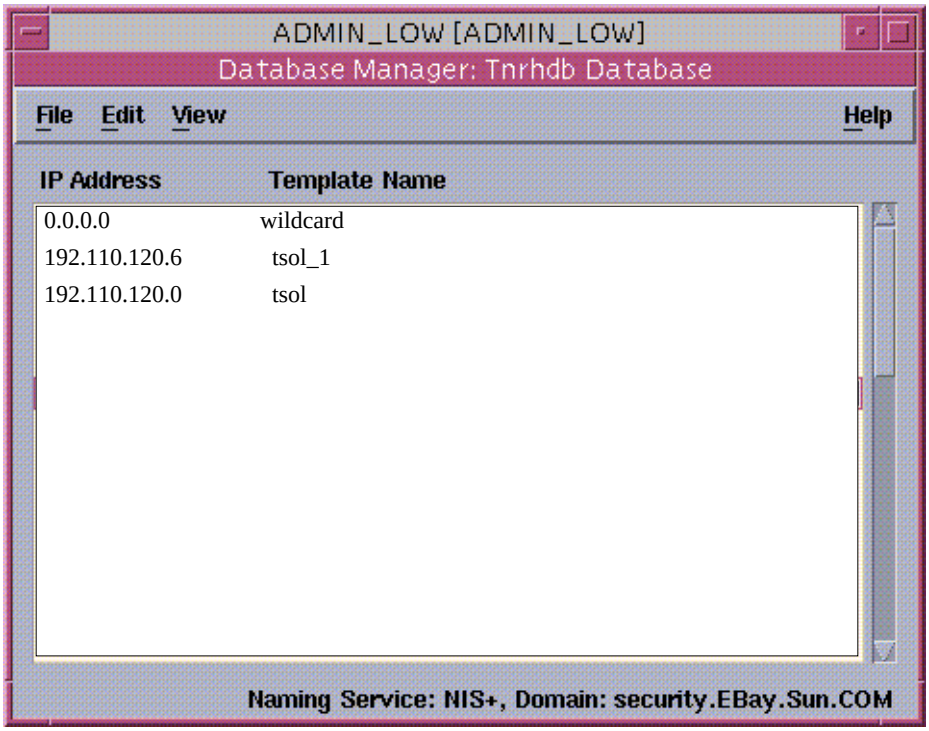



*Figure 10-24* Tnrhdb Fallback Template Entry

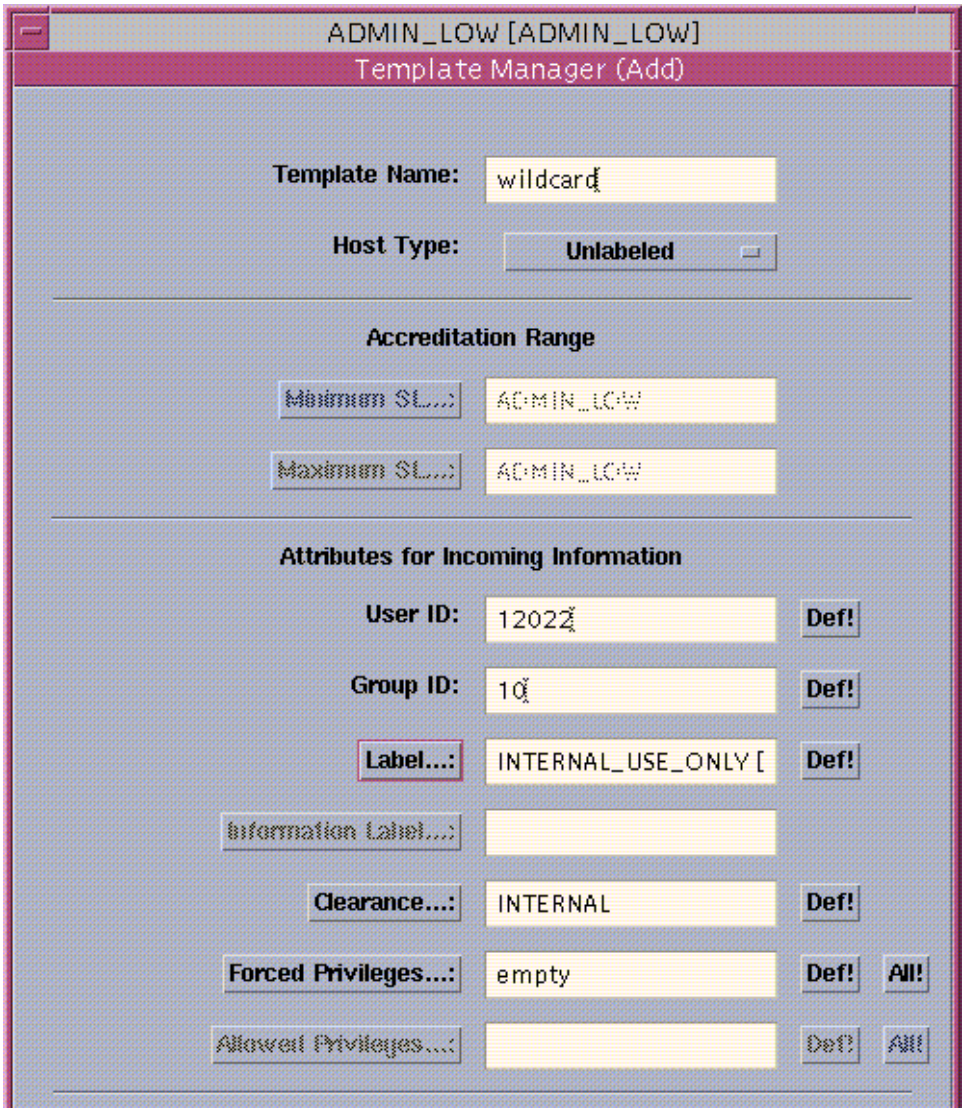



*Figure 10-25* Assigning Some Default Attributes to Communications from Unspecified Hosts

### ▼ To Set an Accreditation Range in a Host Template or Network Interface Entry

1. **Access the Database Manager.**
   See "To Access the Trusted Network Databases from the Database Manager" on page 280, if needed.

**Note –** A separate `tnidb` file resides on each host to specify attributes for that host's own network interfaces, so make sure to choose None in the Naming Service menu in the Database Manager load list when modifying the `tnidb`.

2. **To set a host accreditation range in a template that is assigned to one or more hosts, modify Tnrhtp and specify a Minimum SL and a Maximum SL.**
   a. **Highlight `Tnrhtp` in the Database Manager Load list.**
   b. **To change an existing template, highlight the name of the template and choose Modify.**

**Note –** Several fields in the Template Manager dialog box have a default button (marked Def!) to their right. If the security administrator clicks the default button, a default value is supplied in the field. For example, The default host accreditation range is from `ADMIN_LOW` to `ADMIN_HIGH`.

   c. **Specify the desired sensitivity labels in the Minimum SL and Maximum SL fields, using the label builder dialog boxes that display when you click on the buttons.**

3. **To set a network accreditation range for one or more interfaces on the current host.**
   a. **Highlight `Tnidb` in the Database Manager Load list.**
   b. **To change an existing template, highlight the name of the template whose value you want to set and choose Modify.**
   c. **Enter the desired sensitivity label in the min_SL and max_SL fields.**

**Note –** The default network accreditation range is from `ADMIN_LOW` to `ADMIN_HIGH`.

### ▼ To Configure a Network Interface

1. **If needed, before you start add the physical network interface to the host, following the hardware and software installation steps in the manuals shipped with the interface.**
2. **If there is more than one network interface on the host, do the configuration either for a router or multihomed host as described in the base Solaris** *TCP/IP and Data Communications Administration Guide***.**
3. **If the site security policy requires the default settings for any interfaces on the host be changed, change the entries as described in "To Add a New Entry or Modify an Existing Entry in tnidb(4TSOL)."**
   The default settings are shown inTable 10-6. If the defaults are acceptable, no changes to the `tnidb(4TSOL)` file are needed.

*Table 10-6* Default tnidb Settings

| Minimum SL | Maximum SL | Label (default) | Clearance | User ID | Group ID | Forced Privileges |
|---|---|---|---|---|---|---|
| ADMIN_LOW | ADMIN_HIGH | ADMIN_LOW[ADMIN_HIGH] | ADMIN_HIGH | nobody | nobody | [empty] |

4. **If the name of the new network interface is not already in the `Tnidb` database, add it as described under "To Add a New Entry or Modify an Existing Entry in tnidb(4TSOL)."**
   Figure 10-26 shows the names of the default network interfaces.

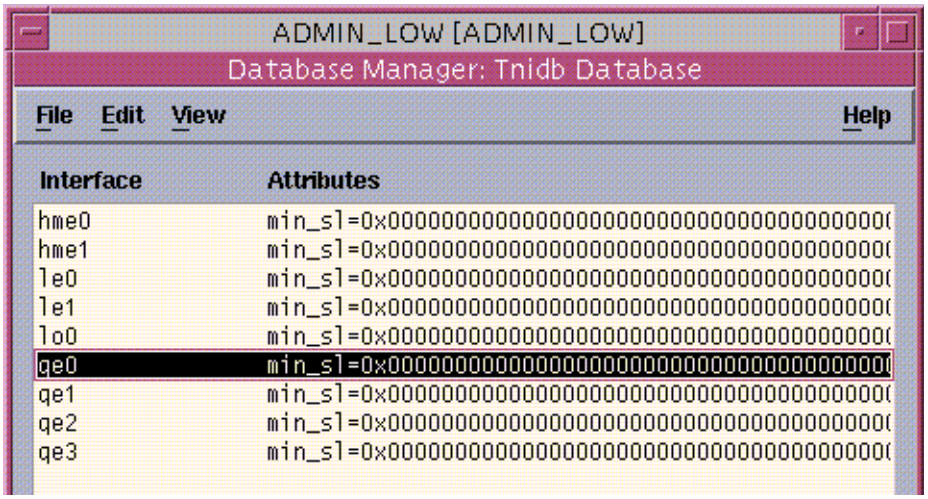



*Figure 10-26* Default Interfaces Listed in the Tnidb Database

### ▼ To Add a New Entry or Modify an Existing Entry in `tnidb(4TSOL)`

1. **Access the Database Manager.**
   See "To Access the Trusted Network Databases from the Database Manager" on page 280 if needed.

2. **If the name of the interface is in the default list, modify the default settings.**

   a. **Highlight the name of the interface and select Modify from the Edit menu.**
   Figure 10-27 on page 293 shows the le0 interface name and the Modify option highlighted on the Edit menu.

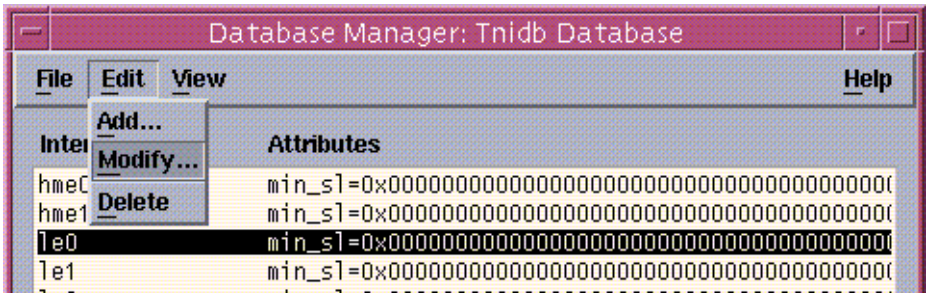



*Figure 10-27* Tnidb Interface le0 Highlighted and the Edit > Modify Option Selected

**b. Modify the settings as desired.**
Figure 10-28 on page 294 shows the default settings for le0.

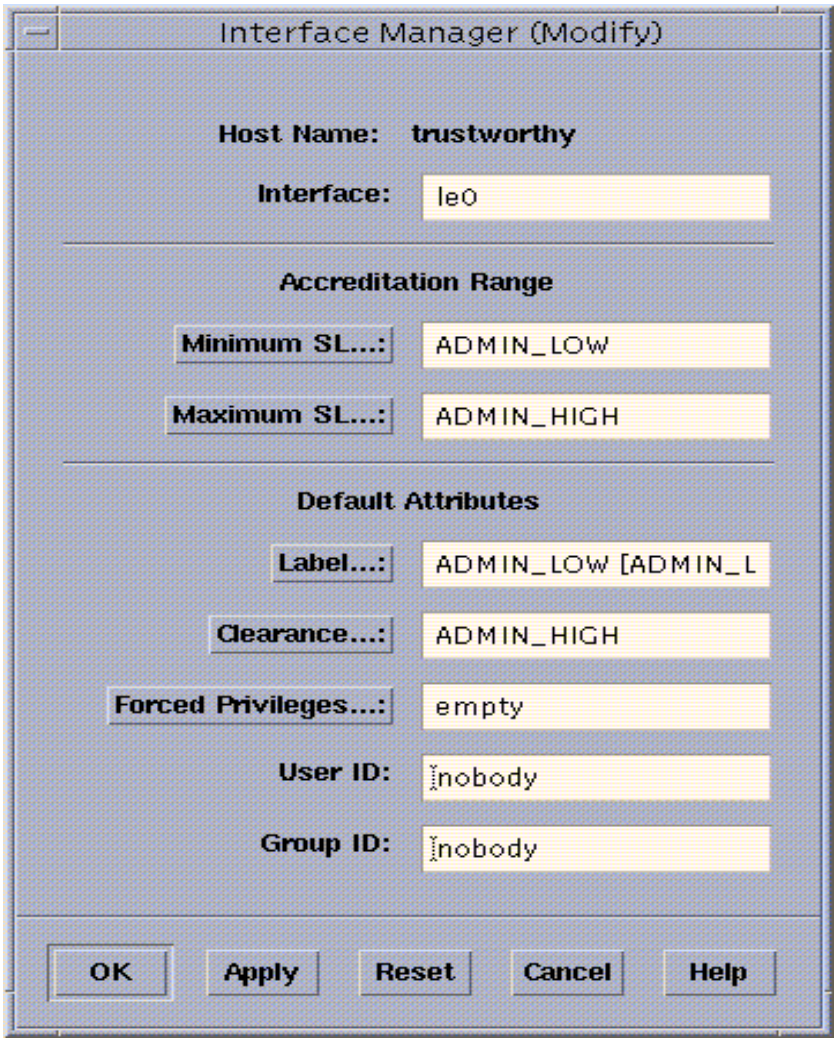



*Figure 10-28* Interface Manager (Modify) Dialog Box

**3. If the name of the interface is not in the default list, add it, and specify the attributes desired.**

**a. Choose Add from the Edit menu.**
Figure 10-29 on page 295 shows the le0 interface name and the Add option highlighted on the Edit menu.

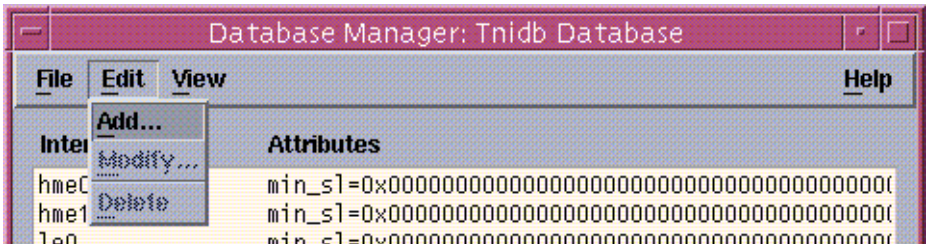



*Figure 10-29* Add Option Selected from the Tnidb Edit Menu

**b. Specify the name of the added interface and its desired attributes.**
Figure 10-30 on page 296 shows the Interface Manager (Add) Dialog Box.

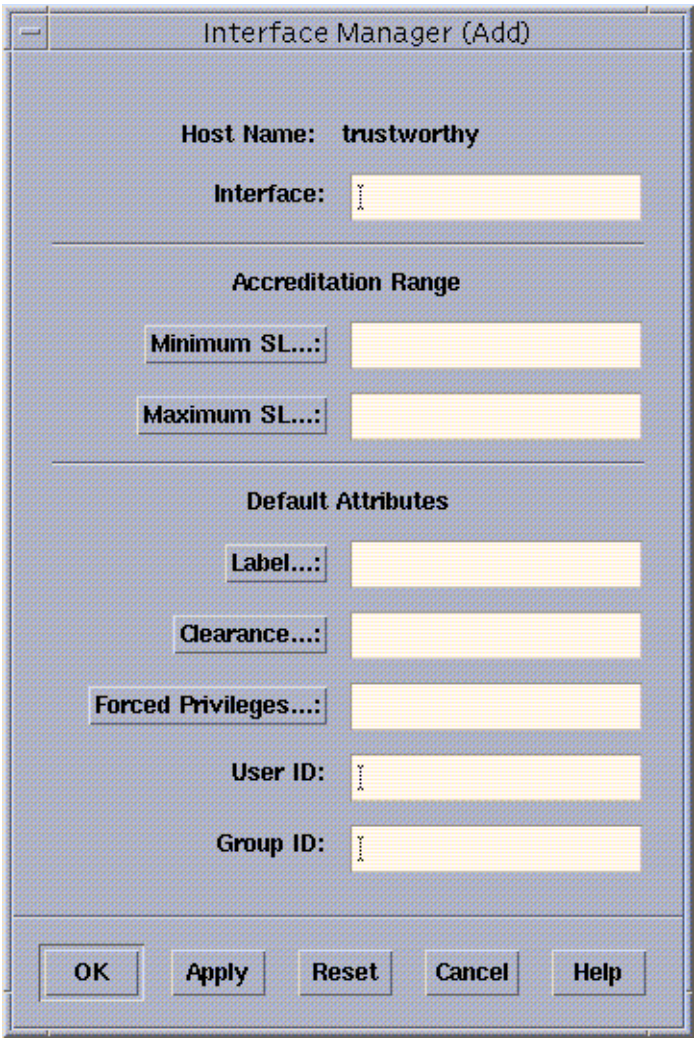



*Figure 10-30* Interface Manager (Add) Dialog Box

### ▼ To Substitute a Valid CIPSO Label for the ADMIN_HIGH Sensitivity Label

1. **Assume the security administrator role.**
2. **Use the Admin Edit action to open the /etc/system file for editing.**

3. **Add a line to set the tsol_admin_high_to_cipso= flag equal to 1.**

```
set tsolsys:tsol_enable_il=1
```

The default, which is not shown in the system file, is set to 0.

## ▼ To Set Up a Simple Default Route for a Network with One Gateway

1. **Assume the system administrator role and access the Set Routes action to create an `/etc/defaultrouter` entry with the hostname of the router.**
   See the `route(1MTSOL)` man page for more about the syntax of `/etc/defaultrouter`.

```
merlot
```

2. **Make sure there is an entry for the gateway(s) in the local `/etc/hosts` file.**

```
129.150.113.36 merlot
```

3. **Make sure there is an entry for the gateway(s) in the local `/etc/security/tsol/tnrhdb` file.**

```
129.150.113.36:tsol
```

## ▼ To Set Up Trusted Routing

**Note –** Do the steps below to set up the Tnrhtp template and assign it to the sending host, to each of the trusted gateways through which the message should pass, and to the destination host.

1. **Access the Database Manager.**
   See “To Access the Trusted Network Databases from the Database Manager” on page 280 if needed.

2. **Double-click on the name of the Tnrhtp database, (as shown in Figure 10-31 on page 298) to bring up the Template Manager, shown in Figure 10-33 on page 300.**

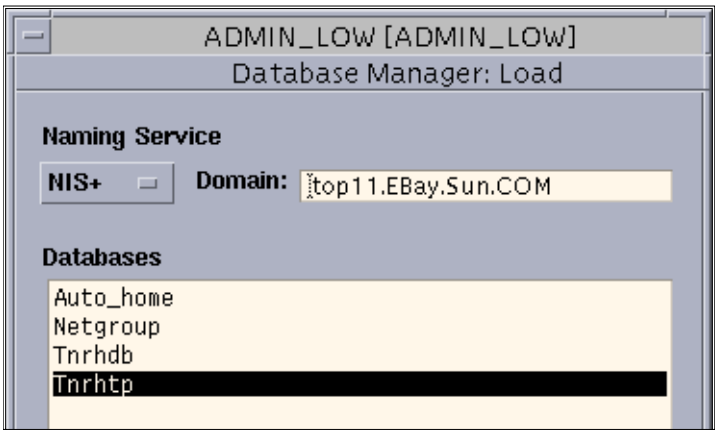



*Figure 10-31* Database Manager: Load List with Tnrhtp Selected

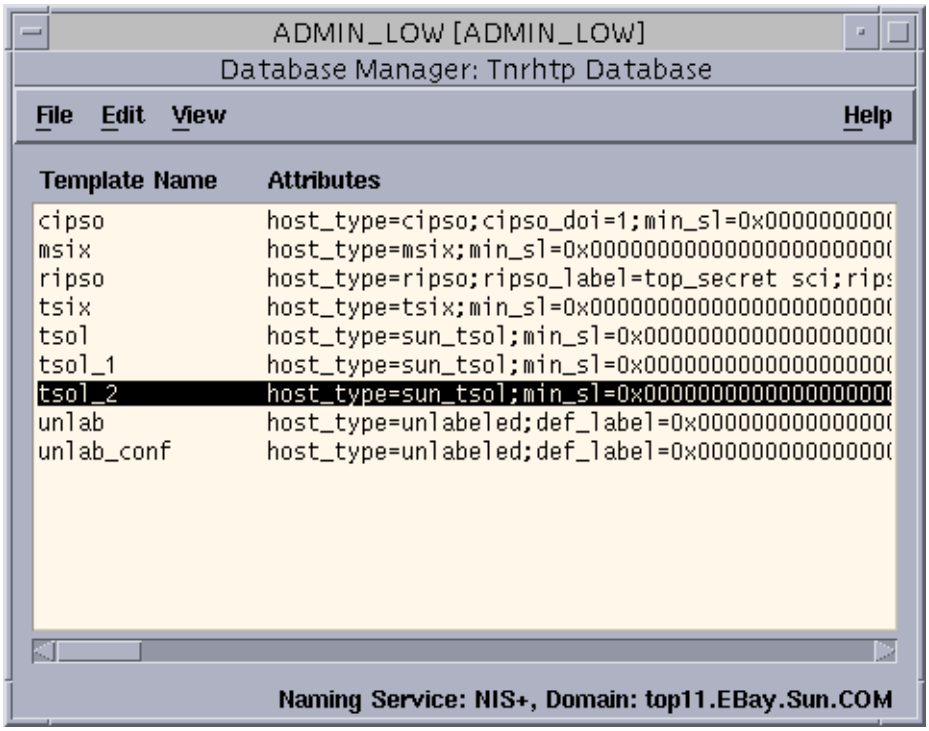



*Figure 10-32* Database Manager: Tnrhtp Database Dialog Box with the tsol_2 Template Name Selected

3. **Double-click on the name of the template to bring up the Template Manager, shown in Figure 10-33 on page 300.**
   The tsol_2 template is selected in Figure 10-32 on page 299.

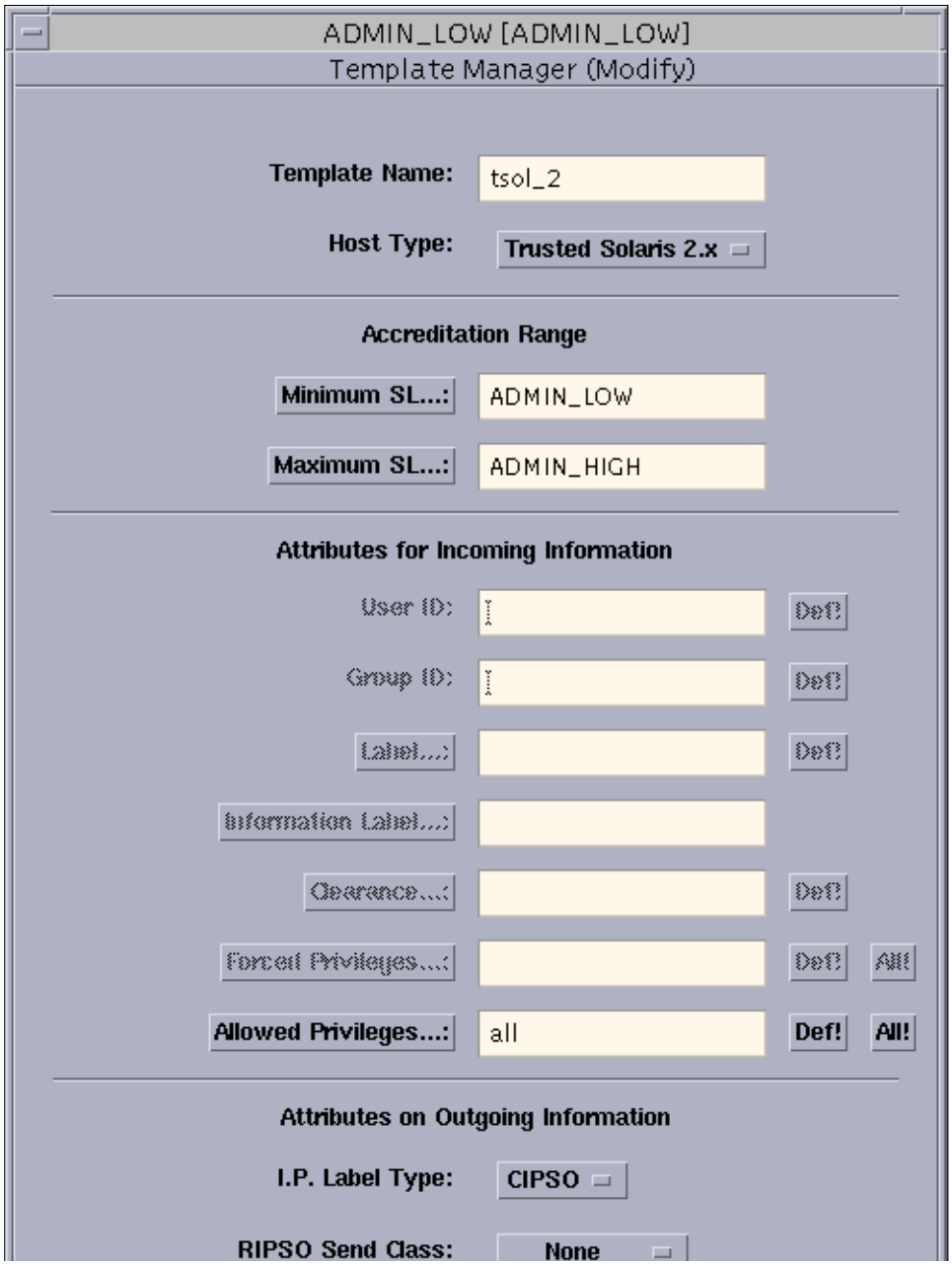



*Figure 10-33* Trusted Network Template Manager Modify Dialog Box

4. **Change any of the fields.**
   In the template for a gateway, you can use either the RIPSO, CIPSO, Trusted Solaris or TSIX host type.

   To use a CIPSO label go to Step 5. To use a RIPSO label, go to Step 6.

5. **To use a CIPSO label, which is derived from the actual sensitivity label on the data, choose CIPSO from the IP Label Type menu and supply a CIPSO DOI.**
   Currently, the only defined CIPSO DOI is 1. Make sure to specify the same CIPSO DOI for the sender, all gateways, and the destination host.

6. **To define a RIPSO label, specify the IP Label Type as RIPSO, select one of the classifications from the RIPSO Send Class menu, choose zero or more of the supported Protection Authority flags from the RIPSO Send PAF menu, and choose one of the supported RIPSO errors in the RIPSO Return PAF menu.**

---

**Note –** Make sure to specify the same RIPSO label and RIPSO error for the sending host, all gateways, and the destination host.

---

7. **Click OK to apply your changes and exit the Template Manager.**

8. **Use the Set Routes action from the System_Admin folder in the Application Manager to open the `/etc/defaultrouter` file.**

9. **List each gateway in the `/etc/defaultrouter` file by name or IP address one line per entry, using either the host's IP address or its host name.**

**Note –** If the hostname is used, the host must also be listed with its IP address in the local `/etc/inet/hosts` file.

The first example shows a gateway listed in `/etc/defaultrouter` by its hostname:

```
routeman
```

The next example shows the same gateway called routeman listed in `/etc/defaultrouter` by its IP address:

```
127.13.104.10
```

# Managing Files and File Systems 11

This chapter gives the background needed to understand how to manage and mount files, directories and file systems and their specify their security attributes in the following sections:


| | |
|---|---|
| *Review of File, Directory, and Filesystem Access Terminology* | *page 305* |
| *Attributes on Files and Directories* | *page 318* |
| *Changing Security Attributes on Files and Directories* | *page 320* |
| *Attributes on File Systems* | *page 323* |
| *Variable Attribute File Systems* | *page 323* |
| *Specifying Variable Attributes on File Systems* | *page 324* |
| *Fixed Attribute File Systems* | *page 325* |
| *Types of File Systems That May Be Mounted in the Trusted Solaris System* | *page 327* |
| *Specifying Mount Time Security Attributes* | *page 330* |
| *Attribute Precedence Rules* | *page 331* |
| *Example of Specifying Security Attributes for a Fixed Attribute File System Mounted from an Unlabeled Host* | *page 333* |
| *Trusted Solaris NFS Mounts* | *page 334* |
| *Trusted Solaris and NFS* | *page 334* |
| *Exporting Directories for Mounting by Other Hosts* | *page 335* |
| *Troubleshooting Mount Failures* | *page 336* |

# 11

This chapter provides the following procedures.


| | |
|---|---|
| *To Change Labels and Privileges on Files and Directories* | *page 336* |
| *To Specify Alternative Security Attributes While Creating a Local File System* | *page 339* |
| *To Set Security Attributes on a Standard File System or Reset Security Attributes for an Existing Trusted Solaris File System* | *page 340* |
| *To Specify Mount-time Security Attributes on the Command Line* | *page 340* |
| *To Specify Mount-time Security Attributes in the Mount Table* | *page 340* |
| *To Mount a TMPFS-type File System Using the Command Line* | *page 343* |
| *To Mount a CDROM (HSFS-type File System) Using the Command Line* | *page 343* |
| *To Trouble Shoot Mount Failures* | *page 344* |


## *Overview of Files, Directories, and File Systems*

The Trusted Solaris system supports the same files and directories, most of the file system types, and all the file system interfaces that are supported by the base Solaris system.

Whenever a process tries to access a file or directory, attributes obtained from various sources are used in making access control decisions. This chapter describes what the various attributes are and where they are obtained and gives the precedence rules that are used to decide between conflicting attributes when they are obtainable from more than one source.

Attributes may be specified:

- At the level of an the individual file or directory within a file system.
- At the level of the file system
- At mount time

# Review of File, Directory, and Filesystem Access Terminology

The terms defined in this section are used in the discussions of how to manage files, directories, and file systems throughout this chapter. These terms and concepts are introduced in the *Trusted Solaris User's Guide* and the *Trusted Solaris Administration Overview*. This review is included here for convenience. You may choose to read through these definitions before going on to the rest of this chapter or to skip to "Attributes on Files and Directories" on page 318 and refer back to this section if you do not understand a term.

Within a definition, special terms used for the first time and terms that need emphasis are *italicized*. Words and phrases in **bold** within the definitions have their own definitions in this section.

## Access Control List

An access control list (ACL) provides a type of **discretionary access control** based on a list of entries that the owner can specify for a file or directory. An access control list can restrict or permit access to any number of individuals and groups, allowing finer-grained control than provided by the standard UNIX **permission bits**.

## Access Permissions

Read, write, and execute or search **permissions** on a file or directory are granted to a process if the **mandatory access control** checks are passed as described in **access policy for files, directories, and file systems**, and if one of the following tests is true.

- If an **ACL** exists for a file or directory, then the following tests are performed in order until one is true, and then the requested access is granted.
  - If the effective UID of the process is equal to the UID of the owner of the file or directory and if the ACL grants the desired type of access to the owner.
  - If the effective UID of the process is in the user list in the ACL and if both the ACL for the owner and the ACL mask grant the desired type of access to the named user.

- If the effective GID or a supplementary GID of the owner of process is equal to the GID of the file or directory and if both the ACL entry for the owner's group and the ACL mask both grant the desired type of access to the owner's group.
- If the effective GID or a supplementary GID of the owner of the process is named in the ACL group list, and if both the ACL entry for the named group and the ACL mask grant the desired type of access to the group.
- If the ACL's *other* entry grants the desired type of access to the owner of the process.

- If an ACL does not exist for the file or directory being accessed, then the following tests are performed in order, and if one of them is passed the desired access is granted.
  - If the effective UID of the process is equal to the UID of the owner of the file or directory and if the owner portion (0700) of the file's permissions is set to allow the desired type of access.
  - If the effective GID is equal to the GID of the file or directory and if the group portion (0070) of the file's permissions is set to allow the desired type of access.If one of the groups in the supplementary group list of the process is equal to the GID of the file or directory and if the group portion (0070) of the file's permissions is set to allow the desired type of access.
  - If the other portion (0007) of the file's permission bits is set to allow the desired type of access to all others.

Otherwise, access is denied, unless the process asserts the appropriate DAC override privilege(s) described in "Access Policy for Files, Directories, and File Systems" and in "Execution Profile Mechanism."

## *Access Policy for Files, Directories, and File Systems*

Because in UNIX systems just about everything (including a spreadsheet, a printer, a letter, a chapter of a book, or a mail message) is handled as a file that is stored in a directory—to do just about anything the user must access files and directories. The conditions for access are described here. (Even though devices are treated as files in the UNIX system, devices have slightly different mandatory access rules than files or directories, and these rules are separately described in this section.) A file, directory, or device may be accessed in three ways:

- The *name* of the file, directory, or device may be *viewed*,
- The *contents* or the *attributes* of the file, directory, or device may be *viewed*, or

- The *contents* or the *attributes* of the file, directory, or device may be *modified*.

In the Trusted Solaris system, each of these types of access is granted or denied based not only on whether the basic UNIX **discretionary access control** checks have been passed but also on whether the **mandatory access control** checks have been passed.

All types of access require that the **sensitivity label** of the **process dominate**s the sensitivity label of all directories in the pathname and that the owner of the process (the person who executed the command) has discretionary search access for each directory in the pathname. View access to the name of the file, directory or device requires only that this part of the check is passed.

For view access (read access) to the contents or attributes of a file or a directory, the process' sensitivity label must dominate the sensitivity label of the file or directory. For view access to the contents of a device (for example, so you can read information stored on a tape in a tape drive), the process' sensitivity label must be equal to the sensitivity label of the device. The owner of the process also must have discretionary read access to the file, directory, or device.

For a process to write into a file or to modify the file's attributes, the sensitivity label of the file must dominate the sensitivity label of the process and must be within the process' clearance, which is set to be the **session clearance**. For a process to write into a directory (create a file), the sensitivity label of the process must equal the sensitivity label of the directory. For a process to write to a device (for example, store information on a tape in a tape drive), the sensitivity label of the process must also equal the sensitivity label of the device. The owner of the process must have discretionary write access to the file, directory, or device.

For each type of failure of a MAC or DAC check, a specific override **privilege** may be applied to the command, depending on the type of access being denied. A privilege can be made available to a command only by the action of a security administrator, because the security administrator must ensure that the user who executes the command is cleared to, or that the command may be trusted to, use the privilege in a trustworthy manner.

These conditions and the listed override privileges apply to any type of access:

- If the sensitivity label of the process does not dominate the sensitivity label of a directory in the pathname, then the process must have the privilege to search up (search a directory whose sensitivity label dominates the sensitivity label of the process), which is *file_mac_search*.
- If the user executing the command does not have discretionary search permission for a directory in the pathname, then the process must have the privilege to override search restrictions when accessing a directory, which is *file_dac_search*.

These conditions and the listed override privileges apply to view (read) access:

- If the sensitivity label of the process does not dominate the sensitivity label of a file or equal the sensitivity label of a directory or device, then the process must have the privilege to override MAC read restrictions, which is *file_mac_read*.
- If the user executing the command does not have discretionary read permission for the file or directory, then the process must have the privilege to override DAC read restrictions, which is *file_dac_read*.

These conditions and the listed override privileges apply to modify (write) access:

- If the sensitivity label of file does not dominate or if the sensitivity label of a directory or device does not equal the sensitivity label of the process, the process must have the privilege that overrides MAC write restrictions, allowing the user to write up and to write above the user's clearance, which is *file_mac_write*.
- If the user executing the command does not have discretionary write permission for the file or directory, then the process must have the privilege to override DAC write restrictions, which is *file_dac_write*.

## *Accreditation Range*

An accreditation range is actually not a range, but a set made up of labels. See “Label Range” on page 311, “User Accreditation Range” on page 317 and “System Accreditation Range” on page 317 for more about the types of label and accreditation ranges in the Trusted Solaris system.

## *Adorned Name*

The text string *.MLD.* is the default adorned name for **multilevel directories** (MLDs). The adornment is alternately called the MLD prefix.

## *CMW Label*

Consists of an **information label** followed by a **sensitivity label** in brackets, in the form: INFORMATION LABEL [SENSITIVITY LABEL].

## *Classification*

The classification is the hierarchical portion of a **sensitivity label**, **information label**, or **clearance**, each of which has only one classification. In a sensitivity label assigned to a file or directory, a classification indicates a relative level of protection based on the sensitivity of the information contained in the file or directory. In a clearance assigned to a user and to processes that execute applications and commands on behalf of the user, a classification indicates a level of trust.

## *Clearance*

The clearance is the upper bound of the set of labels at which a user may work. The lower bound is the minimum label assigned by the security administrator as the initial label. There are two types of clearance, the **user clearance** and the **session clearance**.

## *Compartments*

Compartments are an optional set of words that the site's security administrator may specify to appear in a **sensitivity label**, **information label**, or **clearance**. The compartment represents areas of interest or work groups associated with the labels that contain them and with the files that are assigned the labels and the individuals that work with them.

## *Discretionary Access Control*

Discretionary access control (DAC) is a type of access granted or denied by the owner of a file or directory at the *discretion* of the owner. The Trusted Solaris system provides two kinds of discretionary access controls (DAC): **permission bits** and **access control lists**

## *Dominate*

When any type of label (**sensitivity label**, **information label**, or **clearance**) has a security level equal to or greater than the security level of another label to which it is being compared, the first label is said to dominate the second. The **classification** of the dominant label must equal or be higher than the classification of the second label, and the dominant label must include all the words (**compartments** and **markings**, if present) in the other label. Two equal labels dominate each other. Sensitivity labels are compared for dominance when MAC decisions are being made. See **strictly dominate**.

## *Execution Profile Mechanism*

The execution profile mechanism allows site security administrators to bundle commands, CDE actions, and **security attributes** that may be associated with those commands and actions into an execution profile, which may then be assigned to one or more users depending on the tasks that they need to perform. Security attributes that may be specified in execution profiles include inheritable privileges for commands and actions that may in some cases be used to override discretionary or mandatory **access policy for files and directories and file systems**. See “Overview of Files, Directories, and File Systems” on page 304.

## *Information Label*

An information label is a type of label that signifies the actual security level of the information contained in a file or directory, and which may be used in deciding whether to downgrade the **sensitivity label** of the file or directory, how to physically label information stored on backup media, and how to handle printed output or mail.

## *Information Label Floating*

Information label floating is a conjoining of two **information label**s that occurs when a file or directory with one information label is accessed by a process that has another information label, the resulting information label reflects the combined security level of both information labels.

## *Label*

A label is a security identifier assigned to a file or directory based on the level at which the information being stored in that file or directory should be protected. Depending on how the security administrator has configured the system, users may see the complete **CMW label**, only the **sensitivity label** portion, only the **information label** portion, or no labels at all.

## *Label Range*

A label range is actually a set of **sensitivity label**s assigned to commands, file systems, and allocatable devices, which is specified by designating a maximum label and a minimum label. For commands, the minimum and maximum labels limit the sensitivity labels at which the command may be executed. For file systems, the minimum and maximum labels limit the sensitivity labels at which information may be stored on each file system. Trusted Solaris systems have multilabel file systems configured with a label range from the lowest sensitivity label to the highest sensitivity label. Remote hosts that do not recognize labels are assigned a single sensitivity label, along with any other hosts that the security administrator wishes to restrict to a single label; the label range on a file system mounted from such a host is configured to be restricted to the same sensitivity label as the remote host's sensitivity label. For allocatable devices, the minimum and maximum labels limit the sensitivity labels at which devices may be allocated and restrict the sensitivity labels at which information can be stored or processed using the device. See **network accreditation range**.

## *Mandatory Access Control*

Mandatory access control (MAC) is a type of control based on comparing the **sensitivity label** of a file, directory, or device or another other thing being accessed to the sensitivity label of the **process** that is trying to access it. (The

sensitivity label of the process is the same as the sensitivity label of the workspace where the command was invoked.) Even though directories and devices are managed like files in the UNIX system, the mandatory access control (MAC) rules that apply to directories and devices are different from the MAC rules that apply to files. Before a file may be accessed for writing, MAC checks ensure that the sensitivity label of the file dominates the sensitivity label of the process—a policy called *write up*. A process cannot write to a file whose sensitivity label is higher than the process' **clearance**, which is set to be equal to the session clearance. (The write up policy also includes *write equa***l**.) Before a directory or a device may be accessed for writing, MAC checks ensure that the sensitivity label of the directory or device is equal to the sensitivity label of the process—a policy called *write equal*. Before a file or directory may be accessed for viewing (reading or searching), MAC checks ensure that the sensitivity label of the process dominates the sensitivity label of the file or directory—a policy called *read down*. Before a device may be accessed for viewing, MAC checks ensure that the sensitivity label of the process equals the sensitivity label of the device-a policy called *read equa*l. (The read down policy also includes read equal.)

The rule that applies when a process at one sensitivity label attempts to read or write a file at another sensitivity label is write up, read down (WURD). The rule that applies when a process at one sensitivity label attempts to write a directory at another sensitivity label is write equal, read down. The rule that applies when a process at one sensitivity label attempts to write a device at another sensitivity label is read equal, write equal.

## *Markings*

The codewords, handling caveats, control and release markings and the associated bits that apply to labeled information, markings are only contained in **information label**s.

## *Minimum Label*

For users, the minimum label is the lower bound of the **sensitivity label**s at which a particular user can work, which is specified by the security administrator role while setting the user's account. For the system, the sensitivity label specified in the minimum label field by the security administrator in the `label_encodings` file sets the lower bound for all users.

## *Multilevel Directory*

A multilevel directory is a directory in which information at differing sensitivity labels is maintained in separate subdirectories called single-level directories (SLDs), while appearing to most interfaces to be a single directory under a single name. In the Trusted Solaris system, directories are created as MLDs when they are used by multiple standard applications to store files at varying labels; these include `/tmp`, `/var/mail`, and users' $HOME directories. A user working in an MLD sees only files at the sensitivity label of the user's process. (The exception is if a user has the upgrade file authorization or the downgrade file authorization and is able to upgrade or downgrade a file or directory's sensitivity label after it is created within an MLD. If files or directories have been upgraded, their names are visible, unless the `system(4)` file switch tsol_hide_upgraded_names is changed from the default setting of 0 to 1.)

The **adorned name** of an MLD, also called the MLD prefix, is a file system attribute that may be changed by the system administrator role using `setfsattr(1MTSOL)` with the `-m` option followed by a new prefix. The adornment may also be set on a file system at creation, using `newsecfs` with the `-m` option followed by a prefix. To find out the current MLD prefix, use `getfsattr(1MTSOL)` with the `-m` option. (See also "Working with MLDs" on page 43 in Chapter 2, "Miscellaneous Tasks and Procedures.")

Users can access an MLD two ways: either *without* using the adorned name (such as entering **`cd /tmp`**) or with using the adorned name (as when entering **`cd /.MLD.tmp`**). Interfaces that access directories [such as `cd(1)`, `mkdir(1)`, and `ls(1)`] accept either form. If a file is being created in an MLD in which an SLD at the correct sensitivity label does not already exist, Trusted Solaris creates the SLD and assigns to it the process' sensitivity label.

When an MLD is referred to without using the adorned name. the Trusted Solaris system transparently extends the reference to the single-level directory (SLD) that corresponds to sensitivity label of the window in which the command is invoked; for example, entering **`cd /tmp`** puts the user into the SLD that corresponds to the sensitivity label of the window in which the command is invoked, such as `/.MLD.tmp/.SLD.1`. Use of the adorned name allows programs to refer directly to the MLD instead of to the SLD that has the same sensitivity label as the process. So if the user enters cd /.MLD.tmp puts the user into the MLD, and the user can then list all the SLDs that the current window's sensitivity label dominates.

## Permission Bits

Permission bits are a type of **discretionary access control** in which the owner specifies permissions that are stored as a set of bits to signify who can read, write, or execute a file or directory. Three different sets of permissions are assigned to each file or directory: one set for the owner; one set for all members of the group specified for the file or directory; and one set for all others. See also **access control list**s.

## Privilege

A privilege is a right granted to a process executing a command that allows the command or one or more of its options to bypass some aspect of security policy. A privilege is only granted by a site's security administrator after the command itself or the person using it has been judged to be able to use that privilege in a trustworthy manner. See “Execution Profile Mechanism” on page 310.

## Process

A process is the action that executes a command on behalf of the user who invokes the command. Each process receives a number of security attributes from the user, including the user ID (UID), the group ID (GID), the supplementary group list, and the user's audit ID (AUID). Security attributes received by a process include any privileges available to the command being executed, the process clearance (which is set to be the same as the session clearance), the sensitivity label of the current workspace, and an information label. If the option RESET IL ON EXEC is selected [see `system(4)`], the information label is set to be the lowest viewable label in the system when a new process is started. The information label floats if any information at a higher information label is accessed by the process.

## Security Administrator

In an organization where sensitive information must be protected, the security administrator role is assigned to the person or persons who define and enforce the site's security policy and who are cleared to access all information being processed at the site. In the Trusted Solaris software environment, the security administrator role is an administrative role that is assigned to one or more

individuals who have the proper clearance and whose task is to configure the security attributes of all users and hosts so that the software enforces the site's security policy.

## *Security Attribute*

A security attribute is an attribute used in enforcing the Trusted Solaris security policy. Various sets of security attributes, both from the base Solaris and the Trusted Solaris systems, are assigned to processes, users, files, directories, file systems, hosts on the trusted network, allocatable devices, and other entities. Security attributes for users from the base Solaris system include the user ID (UID), audit ID (AUID), group ID (GID), supplementary group IDs (SGIDs). Security attributes for users from the Trusted Solaris system include the clearance, minimum label (initial label), and any authorizations. An important Trusted Solaris security attribute for files is the CMW label, the sensitivity label portion of which is used in access decisions and the information label portion of which may be used to track the real sensitivity of the information contained in the file. A sensitivity label range security attribute is assigned to file systems, to allocatable devices and to printers. A UID, GID, a label range, and one or more privileges may be associated with commands and CDE actions by security administrators in execution profiles. The mentioned security attributes and a label range called the network accreditation range are assigned to hosts in Trusted Network databases, which are used to control the security of communications in a Trusted Solaris distributed system.The sensitivity labels at which NFS mounts may be performed are limited by the sensitivity labels assigned to the NFS server in the trusted network databases. See also “Attributes on File Systems” on page 323.

## *Security Policy*

In the Trusted Solaris environment, the security policy is the set of DAC, MAC, and information labeling rules that define how information may be accessed. At a customer site, the security policy is the set of rules that define the sensitivity of the information being processed at that site and the measures that are used to protect the information from unauthorized access.

## *Sensitivity Label*

A sensitivity label is a security label assigned to a file or directory or process, which is used to limit access based on the security level of the information contained therein.

## *Session Clearance*

A session clearance is a clearance in effect only during a particular login session, and it is set by the user when starting a session. Each process started during a session has a process clearance equal to the session clearance. The session clearance may be set either to be the same as or lower than the user clearance.

## *Single-level Directory*

A single-level directory (SLD) is a directory within an MLD containing files at only a single sensitivity label. When a user working at a particular sensitivity label changes into an MLD, such as `/tmp`, the user's working directory actually changes to a single-label directory within the MLD, such as /.MLD.tmp/.SLD.1, whose sensitivity label is the same as the sensitivity label at which the user is working. SLD names are created using the .SLD. prefix followed by a number indicating the sequence in which they were created.

## *Strictly Dominate*

When any type of label (sensitivity label, information label, or clearance) has a security level greater than the security level of another label to which it is being compared, the first label strictly dominates the second label. Strict dominance is dominance without equality, which occurs either when the classification of the first label is higher than that of the second label and the first label contains all the compartments in the second label or when the classifications of both labels are the same while the first label contains all the compartments in the second label plus one or more additional compartments.

### *System Accreditation Range*

The system accreditation range is the set of all valid (well-formed) labels created according to the rules defined by each site's security administrator in the label_encodings file, plus the two administrative labels that are used in every Trusted Solaris system, ADMIN_LOW and ADMIN_HIGH.

### *User Accreditation Range*

The user accreditation range is the set of all possible labels at which any normal user may work on the system, as defined by each site's security administrator. The rules for well-formed labels that define the system accreditation range are additionally restricted by the values specified in the ACCREDITATION RANGE section of the site's `label_encodings(4TSOL)` file: the upper bound, the lower bound, the combination constraints and other restrictions.

### *User Clearance*

The user clearance is assigned by the security administrator to set the upper bound of the set of labels at which a particular user may work at any time. The user may decide to accept or further restrict that clearance during any particular login session, when setting the **session clearanc**e after log in.

# 11

## Attributes on Files and Directories

Besides having the attributes that are characteristic of the base Solaris file system, Trusted Solaris files and directories have extended security attributes stored in a shadow inode off of the inode of the object.

Table 11-1 lists the attributes supported in the base Solaris operating system and the additional security attributes required by Trusted Solaris security policy.

*Table 11-1* File and Directory Attributes from the Base and from Trusted Solaris Operating System

| Security Attributes from Solaris Base | Added Trusted Solaris Security Attributes |
|---|---|
| User Id | Sensitivity Label |
| Group Id | Information Label |
| Permission Mode | Forced Privileges |
| Access ACL (optional) | Allowed Privileges |
| Default ACL (optional) | File Attribute Flag |
| | Directory Attribute Flag |

Table 11-2 on page 318 describes the extended attributes.

*Table 11-2* Security Attributes on Files and Directories *(1 of 2)*

| Attribute | Description |
|---|---|
| Sensitivity Label | The sensitivity label of the file or directory |
| Information Label | The information label of the file or directory. For directories, MLDs, and SLDs the information label may be undefined. |

*Table 11-2* Security Attributes on Files and Directories *(2 of 2)*

| Attribute | Description |
|---|---|
| Forced Privileges | The set of privileges that this executable is guaranteed to have available at start of execution. |
| Allowed Privileges | The maximum set of privileges that this executable is allowed to use during its execution. (Because executables that have been edited lose their privileges, limiting the privileges that an executable can use to those in its allowed set provides a protection against Trojan horses, because programs cannot use inheritable privileges if the programs have been edited.) |
| File and Directory Attribute Flag | Flags that indicate directory is an MLD, that a file is a public object. |

Unlike many systems implementing MAC, the Trusted Solaris system does not impose any order on the sensitivity labels of directories in a pathname.

Although the policy is that files and directories can be created only at the same sensitivity label as the containing directory, privileged subjects may create files and directories at any valid sensitivity label and relabel existing files and directories at any valid sensitivity label to create *upgraded objects*. See “To Change Labels and Privileges on Files and Directories” on page 336.

The system can also be configured so that directory entries do not show when a process does not have MAC access to the label of the file or subdirectory, which means that the names of upgraded files and directories are not visible.The Customize Trusted Solaris dialog that displays during installation allows the install team to choose `Hide upgraded names in directories`, which sets a switch in the `/etc/system` file. This setting can be changed after installation by changing the setting of the `tsol_hide_upgraded_name`s switch in the `system` file as described in page 356 of Chapter 13, “Changing Configurable Trusted Solaris Kernel Switches” and rebooting.

Directory name are cleared when a directory is removed, to meet the object reuse requirement that the names of removed directories should no longer be accessible.

Trusted Solaris 2.5 symbolic links have MAC and information label attributes.

The sensitivity label is set at creation and can be changed. The information label is set at creation, and can be changed without causing the process’s information label to float.

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 SLDs within an MLD permits an unprivileged process to combine information from SLDs at its own and lower sensitivity labels. The `mldpwd(1TSOL)` and `mldrealpath(1TSOL)` commands are used to get the name of either the current working directory or of any other MLD with the MLD adornment. Mounting MLDs does not require the use of the adorned name.

For one example of how the security attributes are used to enforce the Trusted Solaris security policy, when a command such as `vi(1)` tries to do something like create (write) a new file in a directory, the security attributes (such as the identity of the user who invoked the command and the sensitivity label of the process) of the process running the command are compared to the corresponding security attributes (in this example the UID and sensitivity label) of the directory where the file is being created, and the operation succeeds or fails based on whether the required MAC and DAC checks are passed. In this example, the user ID would have to meet the DAC requirements specified in the directory's permission bits and in any ACL it might have, and the sensitivity label of the process would have to be dominated by the sensitivity label of the directory.

## *Changing Security Attributes on Files and Directories*

The Trusted Solaris File Manager lets users and administrators change permissions, set privileges and labels on files and directories. Override privileges may be required on the commands or actions if attempted accesses are outside MAC or DAC policy.

### *Changing Labels and Privileges*

The File Manager Selected menu has a Change Labels options to set the sensitivity label and information label, which may also be done on the command line by any account that has the `setlabel(1TSOL)` command in one of its profiles, while the Selected menu's Change Privileges option lets you set forced and allowed privileges on executable files.

The following authorizations are required in order to set privileges and labels, either on the command line or through the File Manager Selected menu options:

- Setting privileges requires the *set file privileges* authorization.
- Upgrading file and directory labels requires the *upgrade file sensitivity label* authorization
- Downgrading file and directory labels requires the *downgrade file sensitivity label* authorization.

See Figure 11-1 for the Selected menu. The options that allow the setting of privileges and labels are new in Trusted Solaris 2.x. See “To Change Labels and Privileges on Files and Directories” on page 336 for how to change labels and permissions.

## *Changing File and Directory Attribute Flags*

The `getfattrflag(1TSOL)` command gets the attribute flag of a file or directory and the `setfattrflag(1TSOL)` command sets the public object flag on a file and sets the MLD flag on a directory.

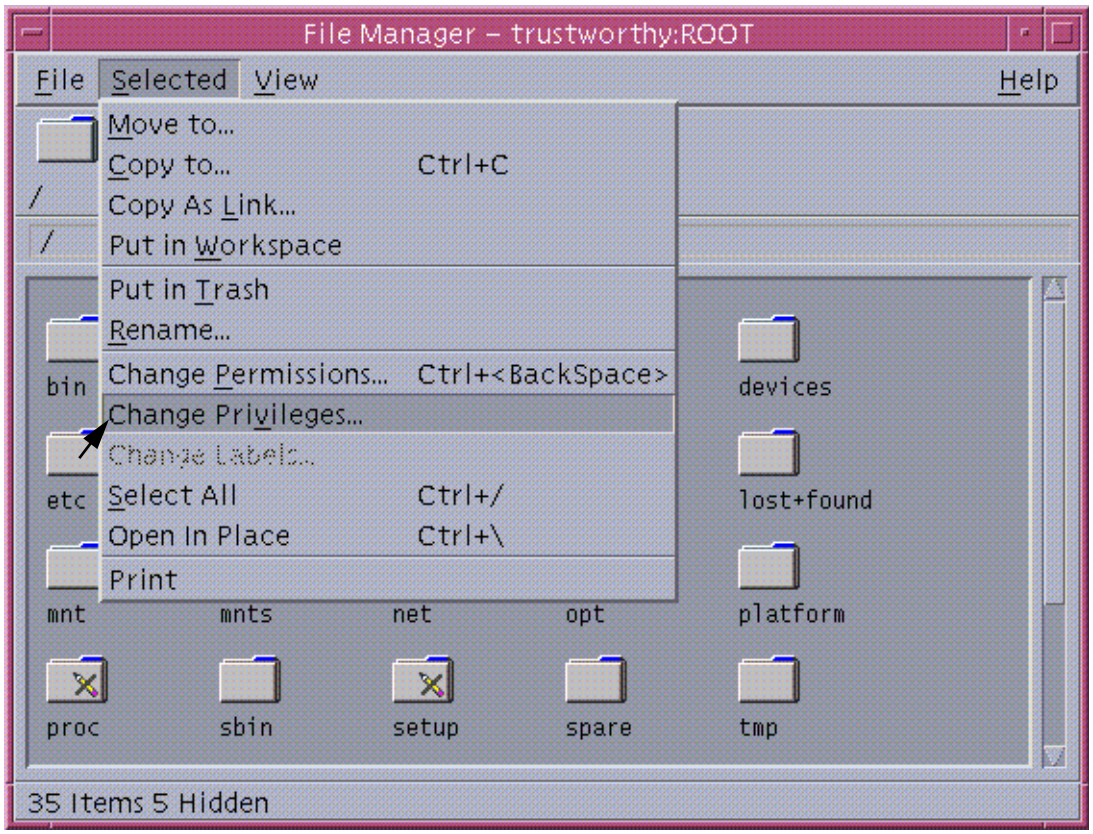



*Figure 11-1* File Manager Selected Menu

# Attributes on File Systems

File systems supported by the Trusted Solaris system are characterized by whether their attributes can be changed or whether they are unchangeable. The two classifications are:

- Variable attribute file systems, and
- Fixed attribute file systems

A file system with the extended Trusted Solaris security attributes is referred to as a *variable attribute* file system or *variable file system*. A file system that does not have the Trusted Solaris security attributes, but has only the standard Solaris attributes or some subset is referred to as a *fixed attribute file system*.

Both variable attribute file systems and fixed attribute file systems can be locally connected to a Trusted Solaris host, even on the same local disk.

## Variable Attribute File Systems

All Trusted Solaris file systems are variable attribute file systems and have the *tsol_attr* flag as one of their extended security attributes, while other types of file systems from other UNIX systems do not have this flag and do not recognize it. On a variable attribute file system with the *tsol_attr* flag:

- The *Trusted Solaris security attributes are present* and
- The security attributes *may be changed.*

Table 11-3 shows the security attributes for variable-attribute file systems, with the default values that are in effect unless they are changed or overridden at mount time by a site's security administrator.

*Table 11-3* Trusted Solaris File System Security Attributes With Defined Settings *(1 of 2)*

| Attribute | Description |
|---|---|
| Filesystem Attribute Flag | The tsol_attr flag. |
| Filesystem MLD prefix | The characters to use for the MLD prefix for MLDs on this file system (default ".MLD."). |
| Filesystem Sensitivity Label Range | The minimum and maximum sensitivity level for files and directories created on this file system (default ADMIN_LOW To ADMIN_HIGH). |

*Table 11-3* Trusted Solaris File System Security Attributes With Defined Settings *(2 of 2)*

| Attribute | Description |
|---|---|
| Filesystem Sensitivity Label | The sensitivity label to infer for all files and directories on this file system that do not have an explicit sensitivity label (default *none*) |
| Filesystem Information Label | The information label to infer for all files and directories on this file system that do not have an explicit information label (default *none*). |
| Filesystem Access ACL | The access ACL to infer for files and directories on this file system that do not have an explicit access ACL (default *none*). See `setfacl(1)` for the format. |
| Filesystem Default ACL | The creation ACL to infer for all files and directories on this file system that do not have an explicit creation ACL (default *none*). |
| Filesystem Forced Privilege Set | The forced privilege set to infer for all executable files on this file system that do not have explicit forced privileges (default *none*). |
| Filesystem Allowed Privilege Set | The allowed privilege set to infer for all executable files on this file system that do not have explicit allowed privileges (default *none*). |

## *Specifying Variable Attributes on File Systems*

All file systems created during Trusted Solaris installation have the *tsol_attr* flag. The default file systems may be used just as they are with the defined set of security attributes. However, when site security policy requires, the security administrator can:

- Use the `getfsattr(1MTSOL)` command to get the security attributes of a file system.
- Use the `newsecfs(1MTSOL)` command to supply security attributes for a new Trusted Solaris file system while creating it.

See “To Specify Alternative Security Attributes While Creating a Local File System” on page 339

- Use `setfsattr(1MTSOL)` to tune an already-existing file system

  See “To Set Security Attributes on a Standard File System or Reset Security Attributes for an Existing Trusted Solaris File System” on page 339.

**Warning –** 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.

## *Fixed Attribute File Systems*

Because they are configured to have a single sensitivity label when mounted on Trusted Solaris hosts, the fixed attribute file systems are also referred to as single-label file systems.

Security administrators may associate a single CMW label (information label with sensitivity label) and a set of attributes with the whole fixed attribute file system and all the files and directories it contains when mounting it. On fixed attribute file systems, neither the information label or the sensitivity label can change as long as an object resides in the file system.

For example, you can NFS-mount a fixed attribute file system called `/spare` from a NFS server called *outside* at PUBLIC[INTERNAL_USE_ONLY] using `mount` with the `-S` option on the command line as shown here:

```
$ mount -F nfs -S slabel=INTERNAL_USE_ONLY\; ilabel=PUBLIC outside:/spare /spare
```

Say that `/spare` contains a file called file. You can not change the sensitivity label or information label of `/spare/file` and its information label can never float. However, if you copy or move `/spare/file` into `/tmp` or `/export/home/secadmin`, for example, you can then change its CMW label.

The security attributes are specified for the mounted fixed file system either on the command line using `mount` with the `-S` option or in the `vfstab_adjunct(4TSOL)` file.

When a fixed attribute file system (such as a file system from a Solaris system) is being mounted, if the system administrator does not specify all the filesystem security attributes, the file system is mounted with security attributes taken from the process mounting the file system. Table 11-4 shows the values used when a fixed attribute file system that does not support an attribute is being mounted and a mount-time value for the attribute has not been supplied either on the command line or in `vfstab_adjunct` entry for the file system.

*Table 11-4* Attributes for Fixed File Systems with Default Values

| Name | Value |
|---|---|
| Filesystem Access ACL | None |
| Filesystem Default ACL | None |
| Filesystem Allowed Privilege Set | Empty |
| Filesystem Audit Preselection Flags | None |
| Filesystem Forced Privilege Set | Empty |
| Filesystem Information Label | Mounting Process's Information Label |
| Filesystem Sensitivity Label | Mounting Process's Information Label |
| Filesystem Label Range | Mounting Process's Information Label |
| Filesystem MLD prefix | .MLD. |
| Filesystem Attribute Flags | None |

**Note –** Empty means that the attribute has all bits off. None means that the attribute has no effect.

## *Types of File Systems That May Be Mounted in the Trusted Solaris System*

Trusted Solaris `mount(1MTSOL)` can be used to mount the following types of file systems:

- FDFS
- HSFS
- LOFS
- NFS
- PCFS
- PROCFS
- TMPFS
- UFS

The CACHEFS filesystem type is not supported.

Multiple `mount_*` mount man pages are available, such as `mount_nfs(1MTSOL)` and `mount_ufs(1MTSOL)`, but there is only one `mount(1MTSOL)` command. The `mount` man page describes security attributes that may be set at mount time and gives the privileges, UID and GID that `mount` needs in order to succeed. See also “Attributes on File Systems” on page 323 and following for more about security attributes.

The options for doing mounts of the various file system types are described in the appropriate mount_* man pages. The `vfstab_adjunct(4TSOL)` man page describes the `/etc/security/tsol/vfstab_adjunct` file, where mount time security options can be entered to be used when file systems specified in the `/etc/vfstab` are being mounted. The differences between fixed attribute and variable file systems are described in “Fixed Attribute File Systems” on page 325.

Table 11-5 lists the various types of mounts with examples and notes.

*Table 11-5* Mount Types, Examples, and Notes *(1 of 2)*

| Type | When Used | Notes |
|---|---|---|
| FDFS | A pseudo file system type that allows a program to access its own file descriptor through the file name space | MAC and DAC isolation are assured because each process can access/see 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 sensitivity label and information label are those of the backing file or directory. This is a fixed attribute file system. |
| HSFS | Mounts a file system from a CD device. | See `mount_hsfs(1M)`. In the Trusted Solaris system, 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 `lofs(7FS)`. In the Trusted Solaris system, the security attributes are identical to those of the underlying file system. |
| NFS | Mounts a file system from a remote NFS server. | See `mount_nfs(1MTSOL)`. See also "Trusted Solaris NFS Mounts" on page 334. NFS mounts can be performed on fixed and variable attribute file systems. |
| PCFS | Mounts DOS file systems from a diskette. | See `pcfs(7FS)`. No extended attributes can be set on this file system type. |

*Table 11-5* Mount Types, Examples, and Notes *(2 of 2)*

| 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 /proc 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 system, 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 `/proc` file, which are imputed from the underlying process's DAC and MAC attributes.If the calling process has the proc owner privilege, then the process can get information at the same sensitivity label about processes not owned by the caller. If the calling process has proc_mac_read privilege, the process can get information about a process that is owned by the caller when the process's sensitivity label dominates that of the caller or is disjoint. The restrictions for modifying are more granular than the ones for reading. See the `proc(4TSOL)` 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 `/tmp` is mounted as a tmpfs. 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 `mount_tmpfs(1MTSOL)`. |
| UFS | Mounts a file system from a local disk | See `mount_ufs(1MTSOL)`. UFS file systems can have fixed mount time attributes assigned or variable attributes assigned at creation or later when the file system is given the tsol_attr flag. See "Variable Attribute File Systems" on page 323. Note: On unlabeled (fixed attribute) file systems, the MLD prefix generally has no useful effect—with the following exception. An mld_prefix should be supplied if another file system that has the tsol_attr flag is being mounted on the unlabeled file system and if the root of that file system is an MLD. If no prefix is supplied; the default is an empty string. |

UFS and NFS mounts are the types of mounts most commonly done.

MLDs are supported only by the following file system types:

- `ufs` -variable (with the tsol_attr attribute)
- `nfs-variable` (with the tsol_attr attribute)
- `lofs`, and
- `tmpfs`

## *Specifying Mount Time Security Attributes*

The `mount` command is extended to allow default values to be specified for privilege sets, ACLs, the sensitivity label, and the information label when a fixed attribute file system is being mounted. For file systems with Trusted Solaris attributes, the mount protocol is extended to retrieve the extended attributes from the file system whether it is local or remote.

The security administrator can specify security attributes at mount time either by using the mount -S option or setting the attributes in the file system's entry in the `vfstab_adjunct(4TSOL)` file. See the `mount(1MTSOL)` man page and:


| | |
|---|---|
| *To Specify Mount-time Security Attributes on the Command Line* | *page 340* |
| *To Specify Mount-time Security Attributes in the Mount Table* | *page 340* |


Allowing the system administrator to specify a set of attributes at mount time:

- Supports mounting of fixed attribute file systems that do not support imbedded security attributes at either the file system or file and directory level
- Permits the security administrator to override already-defined security attributes on variable attribute file systems

When the file system is mounted, any specified mount attributes are first checked against the list of supported attributes for the file system type mounted. If a mount attribute is not supported for the file system being mounted then the mount fails and an error is returned to the caller. For example, on ufs file systems the mount user ID is not supported, because user IDs are supported on ufs file systems. The second check is made on the validity of the attribute value. If the attribute value is not valid then the file system is not mounted and an error will be returned to the caller. For example, the mount forced privilege set must be a subset of the mount allowed privilege set.

A single-label fixed attribute file system allows access to files and directories only at the single sensitivity label range specified when the file system is mounted. Also, sensitivity labels and information labels are not changeable on individual files and directories within the single-label file system. Also, because there is no information labels support, the information labels of all files and directories in a single-label file system are treated as equal to the value specified at `mount` time, do not float and cannot be changed by user action. It is the administrator's responsibility to ensure that the label range and the CMW label specified accurately represent the label of all data.

A file system mounted from a host configured as an unlabeled host supports UIDs, GIDs, and optional ACLs and default ACLs on the files and directories it contains, but has no other security attributes. The security administrator specifies a sensitivity label range that is actually a single label for the file system along with other default security attributes.

When a variable file system with tsol_attr flag is mounted by a Solaris host, the Solaris host ignores the Trusted Solaris security attributes.

## *Attribute Precedence Rules*

Any file or directory's attribute set overrides the file system's attribute set. Any attributes specified when a file system is mounted override the attributes on the underlying file system. Figure 11-2 on page 332 shows the rules.

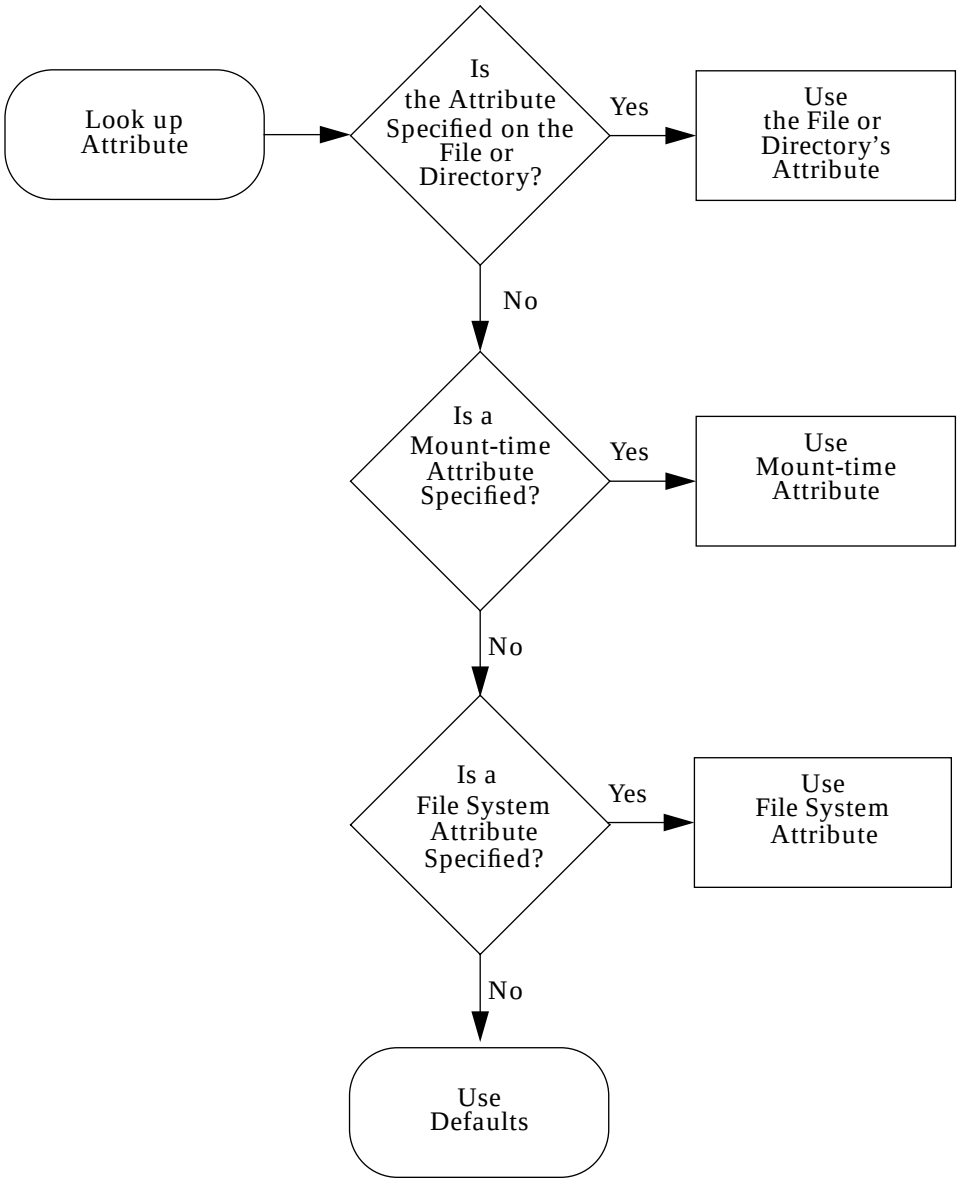



*Figure 11-2* Attribute Precedence Rules

## *Example of Specifying Security Attributes for a Fixed Attribute File System Mounted from an Unlabeled Host*

In this example, the security administrator role has specified the following security attributes for the `/public` file system being mounted from a Solaris host in the `vfstab_adjunct` entry for the file system:

- A minimum sensitivity label and a maximum sensitivity label of PUBLIC
- A default sensitivity label of PUBLIC
- A default information label of PUBLIC
- An allowed privilege set of none
- A forced privilege set of none

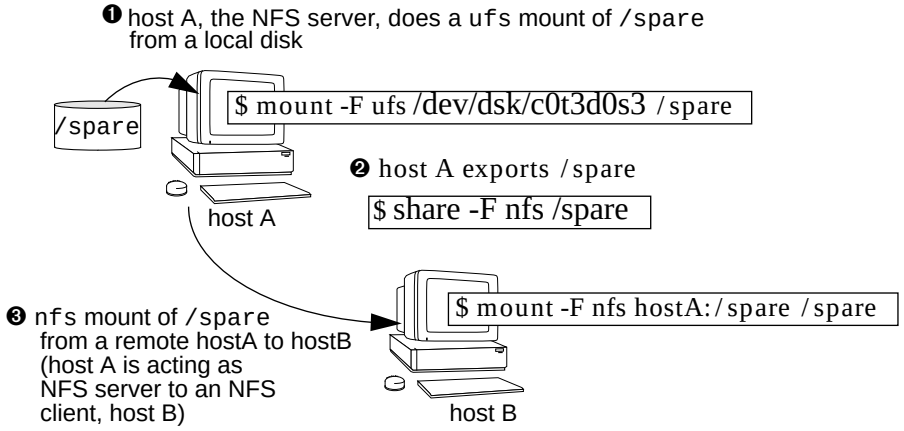
Normal DAC rules apply to attempts to access files and directories in this file system. With the listed defaults set for the file system:

- A user can access a file in that file system (such as `/public/stock.quotes`) only while in a window with a sensitivity label of PUBLIC.
- No action of the user can change the information label or sensitivity label of the file as long as it resides in the `/public` file system.
- If the user copies or moves a file from the `/public` file system to a variable file system with the tsol_attr flag, the file's CMW label changes to that of the process that copies it, and the rest of the file's security attributes can then be changed.

# 11

## Trusted Solaris NFS Mounts

When `mount(1MTSOL)` is invoked to do an NFS-type mount, `mount` is invoked by definition on a Trusted Solaris NFS client to mount a file system from an NFS server. The NFS server may or may not be running the Trusted Solaris operating system. The NFS server has the file system locally-mounted, and the file system can be any local file system type, hsfs, pcfs, tmpfs, ufs, and so forth. The NFS server exports the file system using `share(1M)`. The `share` command and its options are either entered on the command line or in the `/etc/dfs/dfstab` file.




## Trusted Solaris and NFS

Trusted Solaris 2.5 supports both of the Network File System (NFS) protocols supported in the base Solaris operating system and the previous major release of Trusted Solaris:

- NFS Version 2 (V2) (from the Solaris 1.x environment)
- NFS Version 3 (V3) (from the Solaris 2.5 environment)

When a Solaris host exports a file system using the NFS protocol, a Trusted Solaris 2.5 host can use the corresponding NFS protocol version to access the file system at a single label.

A Trusted Solaris host can also use the NFS protocol to export its own file systems to unlabeled client hosts. The unlabeled client ignores the Trusted Solaris security attributes. A file or directory exported to an unlabeled client is *writable* if its sensitivity label equals the sensitivity label associated with the client host in its trusted networking database entries. A file or directory exported to an unlabeled client is *readable* only if its sensitivity label is dominated by the sensitivity label associated with the client host.

To support data sharing with Trusted Solaris 1.x hosts, Trusted Solaris 2.5 partially supports the 1.x version of Trusted NFS. Only the client portion of these protocols is supported so that Trusted Solaris 1.x hosts can export file systems to Trusted Solaris 2.5 hosts. Since some of the extended attributes (such as privileges) are different between the different versions of Trusted Solaris operating system, not all attributes are honored. Specifically, the extended attributes are limited to sensitivity label, information label, and permission bit DAC.[1] No guarantee is made with respect to execution of files from a Trusted Solaris 1.x host. In particular, no privileged programs can be executed from a Trusted Solaris 1.x host.[2]

Any file system being mounted from a NFS server running Solaris 2.4 or earlier versions of Solaris or running Trusted Solaris 1.x needs to be mounted with *vers=2* and *proto=udp* mount options.

The NFS protocol used (whether it is NFS V2/V3, TNFS, TSIG/TNFS) used is independent of the type of the local file system; rather, it depends on the type of the exporting host's operating system. The file system type specified to the `mount` command or in the `vfstab` for remote file systems is always *nfs*.

## *Exporting Directories for Mounting by Other Hosts*

Exporting directories (sharing) for mounting by other hosts is done the same way it is done in the base Solaris system. Two new Trusted Solaris mount options nodev and nopriv may also be used when sharing file systems. See "To Share a Directory for Mounting by Other Hosts" on page 343.

1. No guarantee is made that ACLs are compatibly implemented with Trusted Solaris 1.x.

2. And privileged processes on Trusted Solaris 2.5 may not have their privileges interpreted on a Trusted Solaris 1.x server.

# 11

## 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 the Solaris *System Administration Guide, Volume II*), do the steps in "To Trouble Shoot Mount Failures" on page 344.

## File and File System-related Procedures

### ▼ To Change Labels and Privileges on Files and Directories

1. **Assume the security administrator role.**
2. **Bring up the File Manager and highlight the file whose privileges or label you wish to change.**
3. **To change privileges, choose Change Privileges from the Selected menu.**
   a. **On the File Manager Privileges dialog box shown in Figure 11-4 on page 338, check the button for allowed or forced.**
   b. **Move the desired privileges from the Excluded to the Included list.**
   c. **Click OK.**
4. **To change labels, Choose Change Labels from the Selected menu.**
   a. **On the File Manager Label Builder dialog box shown in Figure 11-4 on page 338, check the button for SL or IL, as needed, and enter a label**
      Do either of Step i or Step ii.
      i. **Type in the text entry field under Update With.**
      ii. **Click the desired classification, compartments or markings, as appropriate.**
   b. **Click OK.**

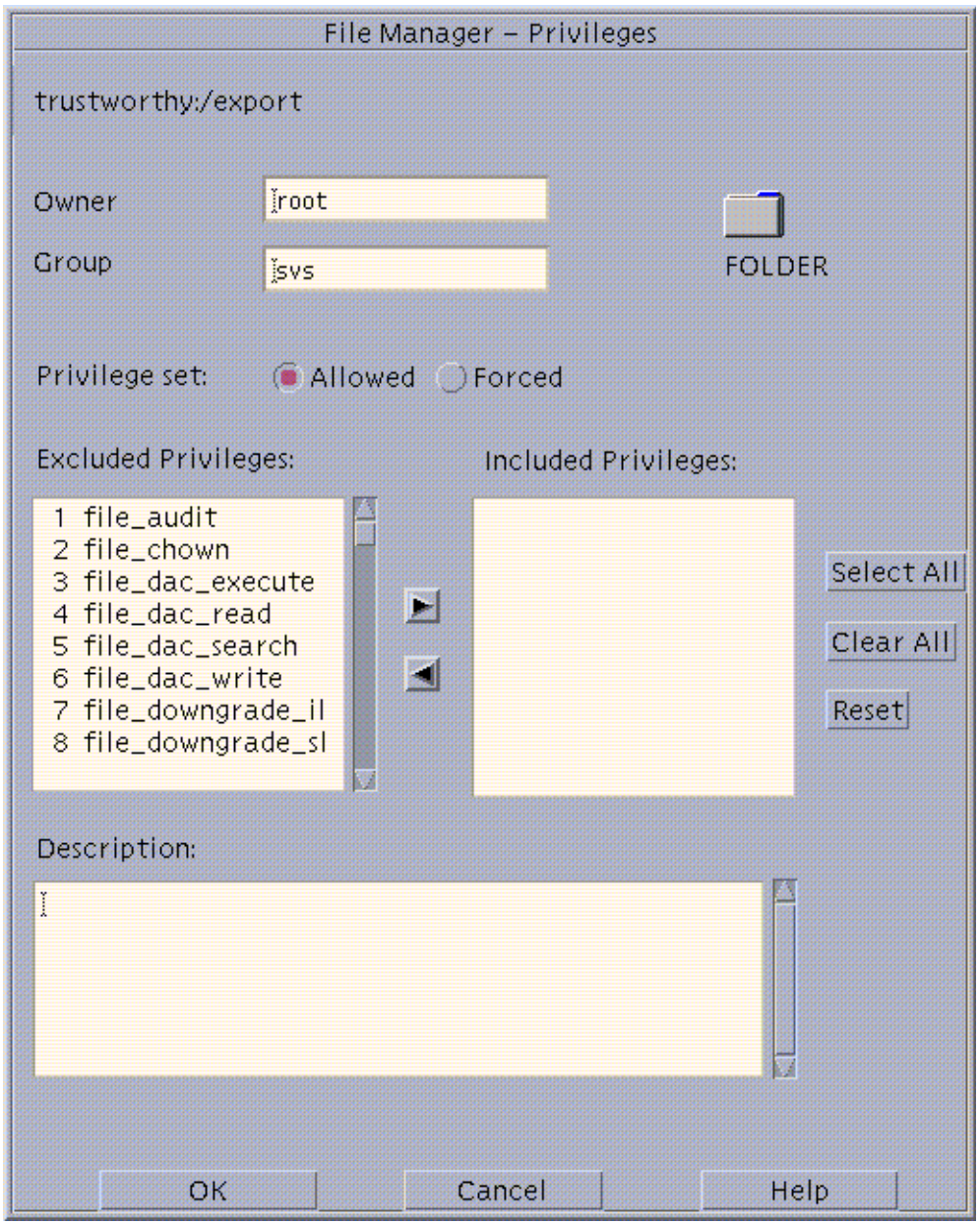



*Figure 11-3* File Manager Privileges Dialog Box

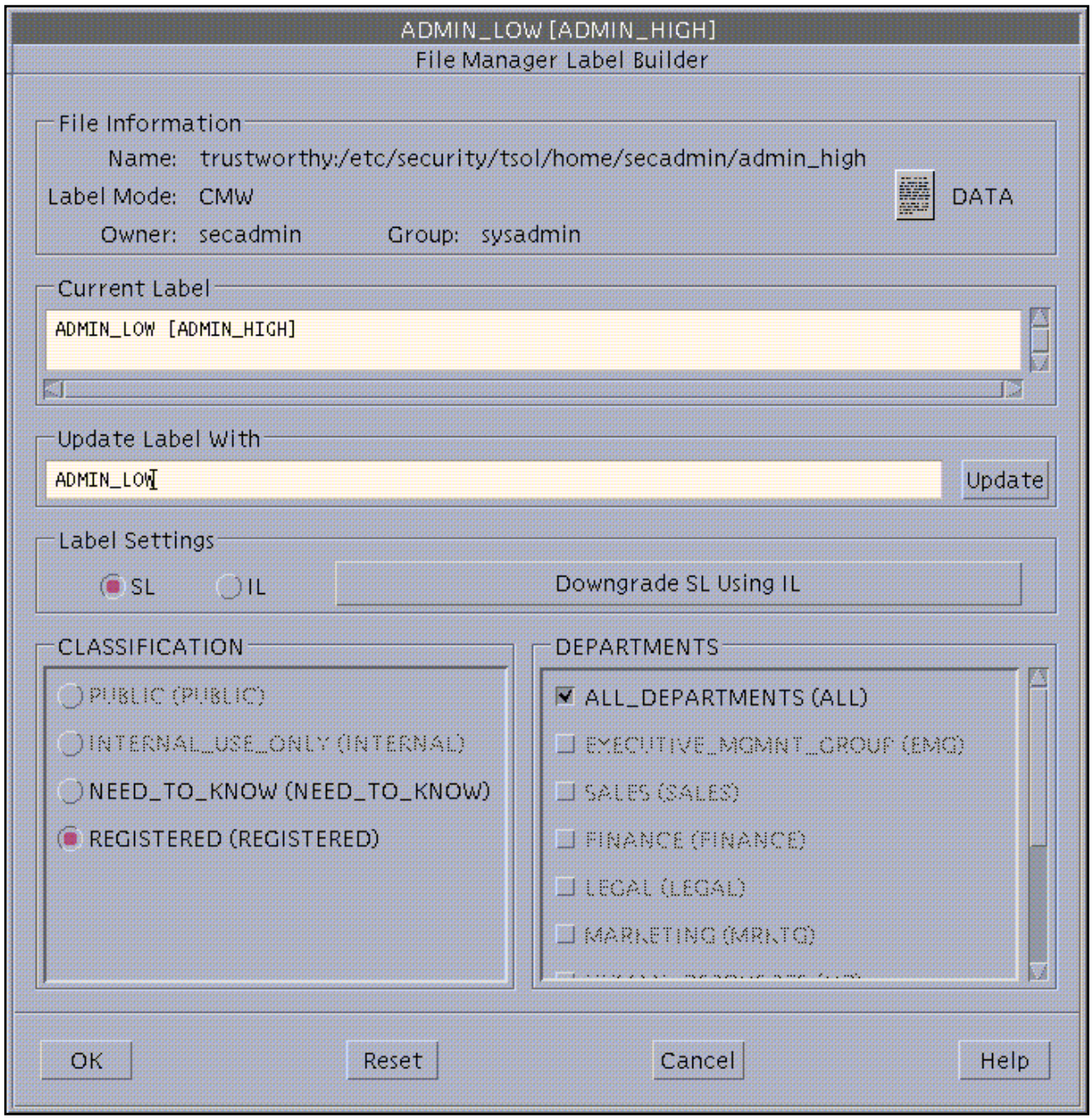



*Figure 11-4* File Manager Label Builder

### ▼ To Specify Alternative Security Attributes While Creating a Local File System

1. **Assume the security administrator role, go to an ADMIN_LOW workspace, and invoke a profile shell [`pfsh(1MTSOL)`].**
2. **Make the mount point directory.**

```
$ mkdir /newpublic
```

3. **Use the Set Mount Points Action to open the `vfstab(4)` file for editing.**
4. **Make an entry for the file system in the `vfstab(4)` file.**

```
/dev/dsk/c0t3d0s3      /dev/rdsk/c0t3d0s3     /newpublic  ufs    2    yes    -
```

5. **Execute the `newsecfs(1MTSOL)` command with the options that specify the desired alternative security attributes, then mount the file system.**

```
$ newsecfs -l Secret;Secret /newspublic
$ mount /newpublic
```

### ▼ To Set Security Attributes on a Standard File System or Reset Security Attributes for an Existing Trusted Solaris File System

1. **Assume the security administrator role and bring up a profile shell [`pfsh(1MTSOL)`] in an ADMIN_LOW workspace.**
2. **Unmount the file system.**

```
$ umount /spublic
```

3. **Use the Set Mount Points Action to open the `vfstab(4)` file for editing.**

4. **Make sure that an entry exists for the file system in the** `vfstab`**(4TSOL) file.**

```
/dev/dsk/c0t3d0s4      /dev/rdsk/c0t3d0s4     /spublic  ufs    2    yes    -
```

5. **Execute the** `setfsattr(1MTSOL)` **command with the appropriate arguments, then remount the file system.**
   The following example sets an sensitivity label range of SECRET to SECRET.

```
$ setfsattr -l Secret;Secret /spublic
$ mount /spublic
```

**Warning –** 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

1. **Assume the system administrator role, and go to an ADMIN_LOW workspace.**
   In a profile shell, enter the mount command, using the -S option followed by an security attributes that you wish to specify.

```
$ mount -F tmpfs -S allowed=all\;mld_prefix=hidden- swap /mnt
```

   The example mounts a tmpfs-type file system, swap, on `/mnt`.

## ▼ To Specify Mount-time Security Attributes in the Mount Table

1. **Assume the security administrator role, and go to an ADMIN_LOW workspace.**
2. **Use the Set Mount Points administrative action to open the** `vfstab(4)` **file for editing.**

3. **Make an entry for the file system in the** `vfstab(4)` **file.**

```
/dev/dsk/c0t3d0s4   /dev/rdsk/c0t3d0s4   /spublic  ufs     2    yes     nodevices,nopriv,nosuid
```

- The `nosuid` mount flag prevents users from using `setuid` in the file system.
- The `nodevices` mount flag disallows opens on device special files.
- The `nopriv` mount flag disallows using privileges on files in the file system.

4. **Save and close the file.**

5. **Use the Set Mount Attributes Action to open the** `vfstab_adjunct(4TSOL)` **file for editing.**

6. **Copy and paste the template entry, and modify the copy.**

The example on Figure 11-5 on page 342 gives the following security attributes to `/spublic`:

- All files in the file system gets an slabel (sensitivity label) of SECRET A so they can only be accessed at that sensitivity label or at a sensitivity label that strictly dominates them.

```
#
#       Yank the following entry and use as a template.
#
#<mount point>; \
#acc_acl=; \
        .
        .
        .
#audit_psa=;
#
#       attributes for an unlabeled filesystem
#
/spublic;\
acc_acl=; def_acl=; mode=; attr_flg=; gid=; uid=; ilabel=;\
slabel="Secret A"; forced=; allowed=; low_range="Secret A";\
hi_range="Secret A"; mld_prefix=; mnt_flg=; audit_psa=;
#
#       attributes for an HSFS file system to mount from a
#       CD-ROM
#
/cdrom;\
acc_acl=; def_acl=; mode=; attr_flg=; gid=; uid=; ilabel=;\
slabel=;
forced=127; allowed=all;
low_range=;hi_range=; mld_prefix=hidden-; mnt_flg=; audit_psa=;
#
#       automatically mounted by /etc/init.d/MOUNTFSYS
#
/tmp;\
acc_acl=; def_acl=; mode=; attr_flg=; gid=; uid=; ilabel=;\
slabel=;
forced=127; allowed=all;
low_range=;hi_range=; mld_prefix=hidden-; mnt_flg=; audit_psa=;
```

*Figure 11-5* Example `vfstab_adjunct` Entries

**7. Save and close the file.**

### ▼ To Share a Directory for Mounting by Other Hosts

1. **Assume the system administrator role in an ADMIN_LOW workspace.**
2. **Use the Share Filesystems action to open the** `/etc/dfs/dfstab` **file for editing.**
3. **Make an entry for the file system you wish to export.**

```
share -F nfs -o nodevices,nopriv,nosuid rw -d "My Home Directory" /export/home/roseanne
```

4. **Run** `shareall` **to tell the NFS daemon** `nfsd(1MTSOL)` **to reread** `/etc/dfs/dfstab`**.**

```
$ shareall
```

### ▼ To Mount a TMPFS-type 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\;mld_prefix=hidden- swap /mnt
```

The example mounts a tmpfs-type file system, swap, on /mnt.

### ▼ To Mount a CDROM (HSFS-type File System) Using the Command Line

**Note –** Automatic mounting of CDROMs does not work in Trusted Solaris.

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 hsfs -o ro -S slabel=p /dev/dsk/c0t6d0s0 /mnt
```

The example mounts an hsfs-type file system at PUBLIC from /dev/dsk/c0t6d0s0 (the CD-ROM device on /mnt.

## ▼ To Trouble Shoot Mount Failures

1. **Make sure that the IP address of the host sharing the file system is in the `tnrhdb(4TSOL)` file on the Trusted Solaris host doing the mounting.**

   If the NIS+ naming service is being used, the IP address of the host sharing the file system must be listed in the `tnrhdb` table in the NIS+ domain; if no naming service is being used, the IP address of the host sharing the file system must be in the `/etc/security/tsol/tnrhdb` file on the Trusted Solaris host doing the mounting.

2. **If the host is not running the Trusted Solaris system, make sure the host has a valid sensitivity label assigned in the `tnrhtp(4TSOL)` template and that the same sensitivity label is being used for the mount, whether the label range is entered after `mount -S` on the command line or in the `/etc/security/tsol/vfstab_adjunct` file.**

3. **If the remote single-label host is newly added to the trusted network databases or its template has been changed since the local host has been rebooted, make sure the entry for the single-label host has been updated with `tnctl -h` *hostname*.**

4. **Ensure that the mount is being done by the administrative role with the `mount` command in one of its execution profiles.**

   In the default configuration, the security administrator specifies the security attributes of mounts while the system administrator takes care of the normal Solaris aspects of mounting.

5. **Make sure to mount any file system that is being mounted from a NFS server running Solaris 2.4 or earlier versions of Solaris or Trusted Solaris 1.x with *vers=2* and *proto=udp* options to `mount(1MTSOL)`.**

# Managing NIS+ 12

To achieve the desired uniformity of user, host, and labels information within a security domain with multiple Trusted Solaris hosts, Sun's naming service (NIS+) is used to distribute most configuration information.

Administering NIS+ is described in the base documentation in:

- *NIS+ Transition Guide*
- *NIS+ and DNS Setup and Configuration Guide*
- *NIS+ and FNS Administration Guide*
- *Name Services Administration Guide*
- *Name Services Configuration Guide*

Setting up the NIS+ master server and NIS+ clients is described in the *Trusted Solaris Installation and Configuration* manual.

This chapter describes the differences in managing NIS+ in a Trusted Solaris system. This chapter includes the following major topics:


| | |
|---|---|
| *Managing Multiple Trusted Solaris Hosts in a Security Domain* | *page 346* |
| *Managing Standalone Trusted Solaris Hosts* | *page 346* |
| *NIS+ Constraint on Using the Root Role to Use Solstice System Administration Tools* | *page 347* |
| *New Trusted Solaris NIS+ Tables and Files Not Administered By NIS+* | *page 347* |
| *Adding Trusted NIS+ Tables* | *page 348* |
| *Adding a New Host and Giving It Credentials* | *page 348* |

# 12

This chapter includes the following procedure.


| | |
|---|---|
| *To Save NIS+ Tables and Restore Them After Reinstalling the Trusted Solaris System* | *page 349* |


## *Managing Multiple Trusted Solaris Hosts in a Security Domain*

A security domain is generally administered as a single NIS+ domain with a single NIS+ master. Multiple security domains may be administered together in a hierarchy of subdomains with multiple NIS+ masters under a single NIS+ root master, which is the server at the top level of a hierarchy of NIS+ domains. There can be only one NIS+ root master. NIS+ replica servers may also be created to provide backup NIS+ query services; the replica is associated with a particular NIS+ master (root or non-root) and responds to NIS+ requests in the event that the primary master is unable to respond.

Configuration files that for one reason or another cannot be administered through NIS+ must be centrally administered and duplicated on individual hosts by other means.

## *Managing Standalone Trusted Solaris Hosts*

Trusted Solaris hosts may or may not be connected to a network with hosts running other operating systems. A standalone Trusted Solaris host may either be configured as its own NIS+ master server or configured with no naming service. If a Trusted Solaris standalone host is configured without a naming service, the configuration information is maintained in `/etc`, `/etc/security`, and `/etc/security/tsol`. The administrative tools in the Trusted Solaris version of Solstice AdminSuite allow the administrative role to specify no naming service so that the information is stored locally.

## *NIS+ Constraint on Using the Root Role to Use Solstice System Administration Tools*

Sites that wish to combine administrative tasks into a single administrative role similar to the root user (superuser) in standard UNIX systems cannot use the root role as the single unified role because:

- The root role cannot be added to the NIS+ admin group, and
- The root role cannot run Solstice from any other host than the NIS+ Master

Trusted Solaris version of the Solstice administration tools allow the system administrator and administrator administrative roles to update files on the NIS+ master. But because the root administrative role can only run Solstice when on the NIS+ master, a combined root role would be similarly constrained.

## *New Trusted Solaris NIS+ Tables and Files Not Administered By NIS+*

New Trusted Solaris NIS+ tables are listed in Table 12-1.

*Table 12-1* New NIS+ Tables

| Map Name | Definition |
|---|---|
| `tsoluser` | Stores Trusted Solaris attributes and execution profiles specified for user and role accounts |
| `tsolprof` | Stores execution profiles. |
| `tnrhdb` | Stores assignments between host and network addresses and templates in `tnrhtp(4TSOL)` |
| `tnrhtp` | Stores templates assigned to hosts and networks in `tnrhdb(4TSOL)` |

Table 12-2 on page 348 shows the configuration files that ordinarily must be the

*Table 12-2* Configuration Files Not Administered Through NIS+

| File Name |
|---|
| `label_encodings(4TSOL)` |
| `system(4)` |

Distributing these files during initial configuration of the Trusted Solaris system is described in the *Trusted Solaris Installation and Configuration* manual. Distributing updated copies of these files if later changes are made is discussed in the *Trusted Solaris Label Administration* manual. See also Chapter 13, "Changing Configurable Trusted Solaris Kernel Switches," in this manual.

## *Adding Trusted NIS+ Tables*

The security administrator can add NIS+ tables with protected data fields by using features of the base NIS+ product. See the NIS+ administration manuals.

## *Adding a New Host and Giving It Credentials*

The system administrator adds a new NIS+ client host using the Host Manager, which configures a host's relationship with the NIS+ master and sets up the host's credentials, while at the same time entering the host's name and IP address into the `hosts` database. The default Secure RPC password assigned to a new host by the Host Manager is *nisplus*.

The Database Manager should be used only to set up communications with a host that is not under local control, because the Database Manager only puts the host's name and IP address into the `hosts` database and does not do any additional configuration. For example, to allow users at a site to use `ftp` to download files from a `ftp` server at MIT, the system administrator would add the MIT host to the `hosts` database using the Database Manager. Since no one at the local site can log into the host, it would not be necessary to set its credentials.

## *NIS+-Related Procedures*

### ▼ To Save NIS+ Tables and Restore Them After Reinstalling the Trusted Solaris System

**1. Dump the NIS+ tables into ASCII files.**

---

**Note –** It is a good idea to dump the NIS+ tables into ASCII files routinely, at least every time you make a change to NIS+.

---

Table 12-3 shows a script the system administrator can create to do the dumps and to create a list of group members for later re-creation of the groups table.

*Table 12-3* nisscript for Dumping NIS+ Tables into ASCII Files *(1 of 2)*

```
#!/bin/sh

# nisscript

# nisplus tables into ascii 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 aliases bootparams ethers group hosts netgroup \

netmasks networks passwd protocols rpc services \

timezone tnrhdb tnrhtp tsolprof tsoluser shadow

do echo $i

/usr/lib/nis/nisaddent -d $i >/var/nis-backup/$i
```

*Table 12-3* nisscript for Dumping NIS+ Tables into ASCII Files *(2 of 2)*

```
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
```

2. **Assume the root role and run the** `nisscript` **at ADMIN_LOW.**
3. **Copy the directory containing the ASCII 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 ASCII dump files in a saved partition, as root at ADMIN_LOW, create a staging directory for the ASCII dumps of NIS+ tables and restore the files from tape or floppy.**
   The screen example illustrates what to do when restoring the ASCII NIS+ files to a `/setup/files` directory from a tape.

```
# cd /setup/files
# tar xv
bootparams
ethers
.
.
.
```

5. **At the appropriate point in** Chapter 5, "Configuring the NIS+ Root Master," in the *Trusted Solaris Installation and Configuration* manual, **recreate the NIS+ environment.**

```
# nisserver -r -d domain-name.
```

   Make sure to include the final period (.) in the domain's name.

6. **As security administrator 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 ASCII dump files reside.**

```
# nispopulate -F -p /setup/files
```

7. **Re-create the NIS+ groups and add members manually from the list of group members created in the `nisscript` shown in Step 1.**
   There is no easy way to recreate the NIS+ groups automatically.

# 12

# Changing Configurable Trusted Solaris Kernel Switches

13

For added configurability, settings of a number of Trusted Solaris kernel switches may be changed by a site's security administrator role. The security administrator role may also set things up so that any changes to the kernel switch settings are passed to each host in the distributed system whenever it boots. This chapter provides the needed background, terms, concepts and procedures for changing the configuration of the Trusted Solaris kernel switches and distributing the changes to all hosts. It includes in the following major topics and procedures sections:


| | |
|---|---|
| *Behaviors Controlled by Configurable Trusted Solaris Kernel Switches* | *page 354* |
| *Needed Terms and Concepts* | *page 354* |
| *How Kernel Switches Are Set and Changed* | *page 357* |
| *To Change Kernel Switch Setting in the /etc/system File* | *page 359* |
| *Distributing Changed Kernel Switch Settings to Hosts Across the Network* | *page 361* |

# 13

## Behaviors Controlled by Configurable Trusted Solaris Kernel Switches

Because they are configurable, the Trusted Solaris kernel switches allow each site to control the following behaviors:

- Whether information labels are displayed at all on the system, and if so,
  - Whether information labels float
  - Whether information labels are reset to 0 upon exec
- Whether the names are displayed of files or subdirectories if their sensitivity labels strictly dominate the sensitivity label of the containing directory
- Allow the use of `runpd(1MTSOL)` to determining which privileges an application needs to run

## Needed Terms and Concepts

This section defines the customer-configurable kernel switches and provides definitions for other related terms.

---

**Note –** Even though the information label-related switches are initially set as described in this section and summarized in Table 13-1 on page 357, the settings may be changed during installation as described in "How Kernel Switches Are Set and Changed" on page 357. The default settings of the switches that are not information label-related may be overridden only by the editing of the `system(4)` file, as described in "To Change Kernel Switch Setting in the /etc/system File" on page 359.

---

### `tsol_admin_high_to_cipso`

As explained in Chapter 10, "Specifying Security Attributes in Trusted Network Databases and Setting Up Routing," when a `tnrhtp` template assigned to the destination host is specified with one of the CIPSO label indicators, the trusted networking software derives a CIPSO label from the message's sensitivity label and inserts the CIPSO label into IP options portion of the message's packets. The Trusted Solaris 2.x ADMIN_HIGH sensitivity label is too big to map to a CIPSO label, so, by default, a message sent to a CIPSO-identified host with the ADMIN_HIGH sensitivity label is always dropped.

This switch must be set to 1 to enable communications with TSIX-type hosts that have the IP Label Field specified as CIPSO.

The security administrator can add the tsol_admin_high_to_cipso switch set equal to 1 in the /etc/system file, which causes the ADMIN_HIGH sensitivity 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.

### tsol_enable_il

This is the master switch for enabling or disabling information label. Some sites do not find information labels useful, and this switch allows them to be turned off completely. If this switch is not enabled, the setting, displaying, and floating of information labels is disabled. This switch is set, and therefore information labels are enabled by default.

### tsol_enable_il_floating

This switch is only looked at if tsol_enable_il=1. This switch is provided for sites that want to use information labels but not allow them to float. If this variable is not enabled, the floating of information labels is disabled. This switch is set, and therefore information label floating is enabled by default.

### tsol_float_sysv_msg_il

This switch is only looked at if both tsol_enable_il=1 and tsol_enable_il_floating=1. If this variable is not enabled, the floating of information labels is disabled for System V Interprocess Communication message queues. This switch is set to 0, and therefore information label floating for this mechanism is disabled by default.

### tsol_float_sysv_sem_il

This switch is only looked at if both tsol_enable_il=1 and if tsol_enable_il_floating=1. If this variable is not enabled, the floating of information labels is disabled for System V Interprocess Communication semaphores. This switch is set to 0, and therefore information label floating for this mechanism is disabled by default.

### tsol_float_sysv_shm_il

This switch is only looked at if both tsol_enable_il=1 and if tsol_enable_il_floating=1. If this variable is not enabled, the floating of information labels is disabled for System V Interprocess Communication shared memory segments. This switch is set to 0, and therefore information label floating for this mechanism is disabled by default.

### tsol_reset_il_on_exec

This switch is only looked at if tsol_enable_il=1. If this switch is not set, the information label of an executed program is set to be the conjunction of both the information label of the calling process, the information label on the program file that is executed, and the information label of any other shared libraries that are read. If this switch is set, the information label of an executed program is reset to ADMIN_LOW at the beginning of execution—which assumes that the arguments and environment passed to the process are not to be labelled with the information label of the caller. When this switch is not set, information labels float even when a file or directory is merely listed, and for this and purely operational reasons, information labels have a tendency to quickly float to the highest information label and thus lose their significance. This switch is set, and therefore information labels are reset to ADMIN_LOW by default.

## *Upgraded Names*

Actions by users with the upgrade file sensitivity label authorization and by processes with the file_mac_write and file_upgrade_sl privileges may either create a new file or subdirectory or relabel an existing file or subdirectory at an sensitivity label that strictly dominates the sensitivity label of the containing directory; these files and subdirectories are said to be upgraded and the names of the upgraded files and subdirectories are referred to as *upgraded names.*

### tsol_hide_upgraded_name*s*

At sites that consider *upgraded names* to be sensitive information, the tsol_hide_upgraded_names kernel switch allows the security administrator role to configure the system so that upgraded names are hidden. Setting this flag prevents upgraded file names from being returned with `getdents(2)`.

Because all directory entries must be examined before the results are returned to the calling process, there is a performance penalty. This switch is not set, and therefore upgraded names display by default.

### tsol_privs_debug

Allows the administrative use of `runpd(1MTSOL)` to characterize a program's use of privilege. Requires additional setup; for the complete procedure, see Chapter 16, "Adding Software" under "To Find Out Which Privileges an Application Needs" on page 473. After the application(s) have been privileged debugged, this variable must be reset and the machine rebooted. This switch is not set, and therefore privilege debugging is disabled by default.

## *How Kernel Switches Are Set and Changed*

Table 13-1 summarizes information about each of the Trusted Solaris kernel switches that are available for setting by customers. The kernel switches are initially set to the defaults shown in Table 13-1 by the Trusted Solaris installation process.

*Table 13-1* Default Kernel Switch Settings and Values Definitions

| Switch Name | Value Definitions | Default Settings |
|---|---|---|
| `tsol_admin_high_to_cipso` | 0 ILs are disabled<br>1 ILs are enabled | `tsol_admin_high_to_cipso=0` |
| `tsol_enable_il` | 0 ILs are disabled<br>1 ILs are enabled | `tsol_enable_il=1` |
| `tsol_enable__il_floating` | 0 IL floating is disabled<br>1 IL floating is enabled | `tsol_enable_il_il_floating=0` |
| `tsol_float_sysv_msg_il` | 0 IL floating is disabled on System V message queues<br>1 IL floating is enabled on System V message queues | `tsol_float_sysv_msg_il=0` |
| `tsol_float_sysv_sem_il` | 0 IL floating is disabled on System V semaphores<br>1 IL floating is enabled on System V semaphores | `tsol_float_sysv_mem_il=0` |

*Table 13-1* Default Kernel Switch Settings and Values Definitions

| Switch Name | Value Definitions | Default Settings |
|---|---|---|
| `tsol_float_sysv_shm_il` | 0 IL floating is disabled on System V shared memory segments<br>1 IL floating is enabled on System V shared memory segments | `tsol_float_sysv_shm_il=0` |
| `tsol_str_linkb` | 0 linkb does not drop a streams msg if an attempt is made to link it to another message with different streams attributes<br>1 linkb drops a streams msg if an attempt is made to link it to another message with different streams attributes | `tsol_str_linkb=1` |
| `tsol_hide_upgraded_names` | 0 Show upgraded names<br>1 Hide upgraded names | `tsol_hide_upgraded_names=0` |
| `tsol_privs_debug` | 0 Disable debugging<br>1 Enable debugging | `tsol_privs_debug=0` |
| `tsol_reset_il_on_exec` | 0 Set IL to the conjunction of the ILs of the calling process, the exec'd process, and any shared libraries.<br>1 Reset IL to ADMIN_LOW upon exec. | `tsol_reset_il_on_exec=1` |

During installation, the install team may accept or reset the default information label-related kernel switches in the Customize Trusted Solaris Configuration dialog box, shown in Figure 13-1 on page 359. See also the *Trusted Solaris Installation and Configuration* manual. Any changes to the default settings are made

in the kernel and recorded in the `/etc/system` file, which is then read when the system is booted.

After installation, any of the kernel switches shown in Table 13-1 may be changed by the security administrator, by modifying the `system(4)` file, as described in "To Change Kernel Switch Setting in the /etc/system File" on page 359.

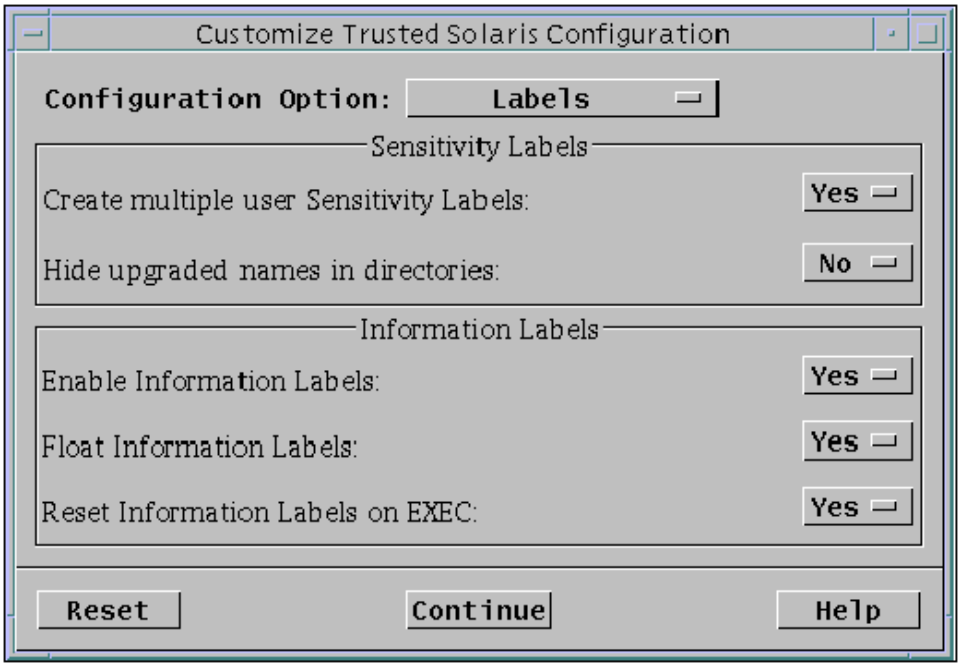



*Figure 13-1* Label Configuration Defaults

## ▼ To Change Kernel Switch Setting in the `/etc/system` File

1. **As security administrator, use the Admin Editor action from the System_Admin folder in the Application Manager to open `/etc/system` for editing.**
2. **To turn on or off the variable that substitutes a CIPSO label for the ADMIN_HIGH sensitivity label, find `tsol_admin_high_to_cipso=` and set the value to 1 (on) or 0 (off).**

3. **To turn on or off the variable for enabling information labels, find** `tsol_enable_il=` **and set the value to 1 (on) or 0 (off).**
   Because the switches for information label floating and for the reset of the information label when a new program is executed are looked at only when the switch shown in Step 3 is set to 1, do either of Step a or Step b only if you have enabled information labels.

   a. **To turn on or off the variable for enabling information label floating, find** `tsol_enable_il_floating=` **and set the value to 1 (on) or 0 (off).**

      i. **To turn on or off the variable for enabling information label floating on System V message queues, find** `tsol_float_sysv_msg_il=` **and set the value to 1 (on) or 0 (off).**

      ii. **To turn on or off the variable for enabling information label floating on System V semaphores, find** `tsol_float_sysv_sem_il=` **and set the value to 1 (on) or 0 (off).**

      iii. **To turn on or off the variable for enabling information label floating on System V shared memory segments, find** `tsol_float_sysv_shm_il=` **and set the value to 1 (on) or 0 (off).**

   b. **To turn on or off the variable for resetting information labels on** `exec`**, find** `tsol_reset_il_on_exec=` **and set the value to 1 (on) or 0 (off).**

4. **To turn on or off the variable for hiding the names of files whose sensitivity labels have been upgraded, find** `tsol_hide_upgraded_names=` **and set the value to 1 (on) or 0 (off).**

5. **To turn on or off the variable for allowing privilege debugging, find** `tsol_privs_debug=` **and set the value to 1 (on) or 0 (off).**

6. **Save and close the file.**

7. **Reboot.**

8. **To distribute the changes to all hosts in the distributed system, follow the steps in "Distributing Changed Kernel Switch Settings to Hosts Across the Network" on page 361 of** *Trusted Solaris Label Administration*
   The security administrator can set up the boot server so that any changes to the `/etc/system` file are passed to each host in the distributed system from a file consulted at boot.

## *Distributing Changed Kernel Switch Settings to Hosts Across the Network*

Modifications seldom need to be made to `system(4)` file after an site has been installed and configured, except for changing the switch that enables privilege debugging, and enabling privilege debugging is usually done on a single host. The security administrator can use the `rdist(1)` command to automatically distribute identical copies of the file to all machines in the distributed system.

### ▼ To Remotely Distribute the `system` File

**Note –** Make sure that every host has only a plus in the `hosts.equiv` file, and that there are no entries in either the `/.rlogin` or in any `$HOME/.rlogin` files.

1. **As security administrator in an ADMIN_LOW workspace, use the Admin Edit action to open and modify a `distfile` to copy the configuration files from a master directory.**
   The example shows a sample `distfile` that is set up to tell `rdist` to copy the `label_encodings` and `system` file to all the listed hosts in the distributed system.

```
# #
HOSTS = ( machiavelli muckraker mugwump diehard warhorse )
FILES = ( /etc/security/tsol/label_encodings /etc/system )

${FILES} -> ${HOSTS}
    install ;
```

2. **Run the `rdist` command.**
   You can either run `rdist` in the same directory as the `distfile` or use `rdist` with the `-f` option followed by the full pathname of a file with some other name.

```
$ rdist \* OR *\
$ rdist -f /home/machiavelli/jedgar/label_encodings.master/distfile.sample
```

See also the `rdist(1)` and `hosts.equiv(4)` man pages.

3. **Reboot each machine.**

# Managing Printing 14

Standard Solaris print utilities and databases and the Solstice Printer Manager have been modified to meet Trusted Solaris requirements. The system administrator and the security administrator role both share printer administration duties, which require an understanding of printer administration concepts, as described in the Solaris *System Administration Guide*, and of how to use the Solstice Print Manager, as described in the *Solstice AdminSuite 2.1 Print Administration Guide*.

This chapter provides background information on the following topics for understanding and managing the unique aspects of printing in the Trusted Solaris environment:


| | |
|---|---|
| *How Trusted Solaris Features Address Information Labeling and Access Control for Printers* | *page 365* |
| *Assigning Labels to Print Jobs* | *page 366* |
| *Using a Label Range on Printers to Control Which Jobs Can Print* | *page 367* |
| *Printing of Labels on Printer Output* | *page 368* |
| *Labels, Job Numbers, and Handling Information Printed on Banner and Trailer Pages* | *page 370* |
| *Supported Printers and Types of Output* | *page 378* |
| *Configuring Printers* | *page 381* |
| *Modified Utilities and Man Pages* | *page 382* |
| *Authorizations to Bypass Printing Defaults* | *page 383* |

# 14

This chapter also provides the following procedures:


| | |
|---|---|
| *To Access the Printer Manager* | *page 384* |
| *To Install a Printer on a Print Server* | *page 386* |
| *To Configure a Restricted Label Range for a Printer Using the Device Manager* | *page 392* |
| *To Add Access to a Remote Printer* | *page 394* |
| *To Specify SLs to Print Instead of ILs on Body Pages* | *page 396* |
| *To Allow Some Users to Print Jobs Without Banners and Trailers* | *page 397* |
| *To Suppress the Printing of Page Labels on All Print Jobs* | *page 398* |
| *To Allow Some Users to Print Jobs Without Page Labels* | *page 398* |
| *To Set Up Automatic Printing of Publicly-readable Jobs Without Labels* | *page 399* |


## Needed Terms

### Banner/Trailer Pages

In the base Solaris printing system, jobs normally print with an optional banner page that contains information on, for example:

- Who submitted the job
- The time of printing, and
- The name of the host the job was submitted from

In the Trusted Solaris system, every job by default has a trailer page paired with a banner page. The banner and trailer pages contain additional security-related information about the printed output. The print service guarantees the content and accuracy of the information contained in the banner/trailer pages. Users may be allowed to suppress the printing of banner and trailer pages with a special authorization.

### Body Pages

The body pages for a print job are the pages that contain the data submitted by a user for printing. The body pages are sandwiched between the banner and trailer pages, and in the Trusted Solaris system, the body pages are printed with labels at the top and bottom of each page. Users may be allowed to suppress the labels on body pages with a special authorization.

## *How Trusted Solaris Features Address Information Labeling and Access Control for Printers*

By default, the Trusted Solaris printing subsystem provides the following features:

- Banner/trailer pages on all print jobs have the print job's sensitivity label, information label, and a unique job ID, along with site-specified special handling instructions that are based on the label of the print job
- The print job's information label is printed on the top and bottom of body pages
- Security-sensitive information about the print job and the print data itself are protected
- Status information provided to users is controlled by MAC and DAC
- Each printer can be configured with a restricted sensitivity label range to control which jobs it prints

The banner and trailer pages and the printing of labels at the top and bottom of body pages can be suppressed either by command line options used by authorized user or role accounts or by administrative action that turn off these features for everyone.

The system's MAC and DAC policies are enforced upon the data contained in each print request. Enforcement is from the point at which the print request is submitted until the requested data has been printed on a physical page. Attempts to subvert or circumvent the protection provisions are detectable and generate an audit trail.

The handling of printer output is controlled by each site according to its policy and procedures.

All labeling of printer output from hosts running the Trusted Solaris operating system is *automatically done* according to the site's requirements. Access to printers is controlled by comparing the label of the print job to the label range of the printer.

# 14

## *Assigning Labels to Print Jobs*

Each print job is automatically assigned a sensitivity label that corresponds to the sensitivity label at which the user is working. Figure 14-1 shows an employee reading email at a sensitivity label of `INTERNAL_USE_ONLY`. When he sends the email to the printer by selecting the Print option from the Message menu, the print job is automatically assigned the sensitivity label `INTERNAL_USE_ONLY`.

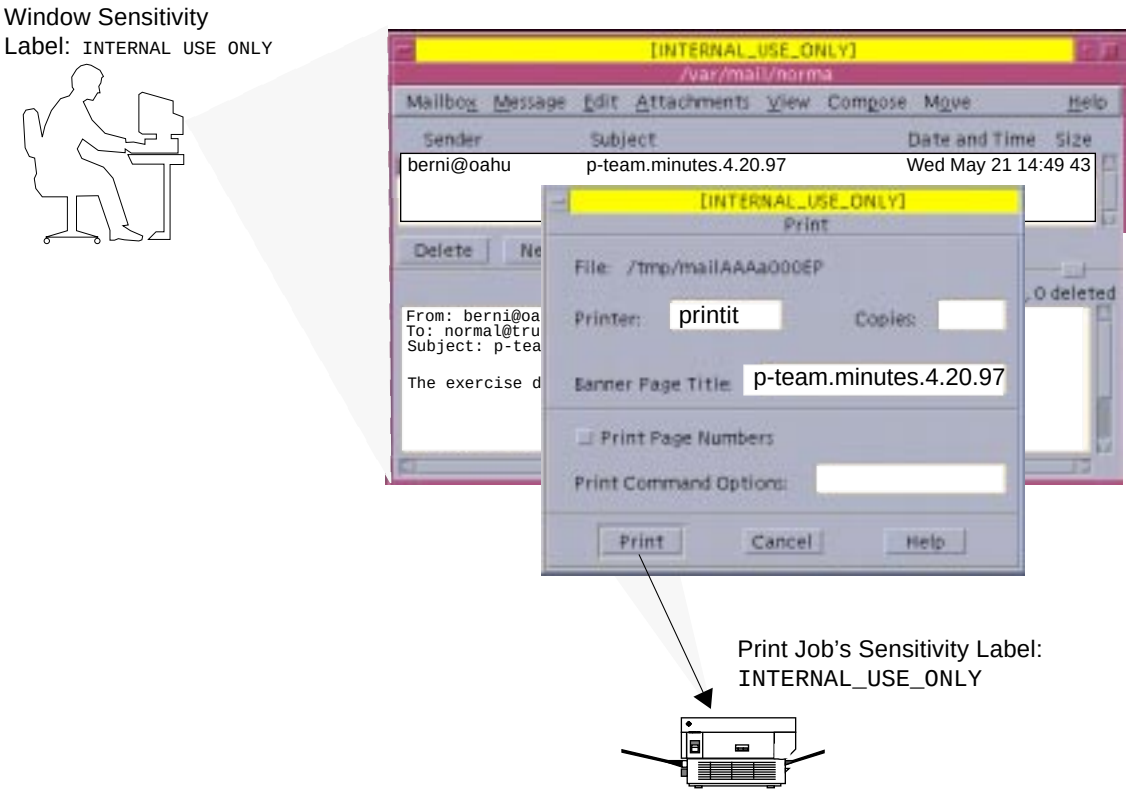



*Figure 14-1* Automatic Labeling of Print Jobs

## *Using a Label Range on Printers to Control Which Jobs Can Print*

Printers may be configured to print only jobs whose labels are within a restricted label range. Email notifies the sender when a job does not go through.

The job that was sent to print at the INTERNAL_USE_ONLY label in Figure 14-1 on page 366 is accepted by the legal department's printer, because it is set up to print jobs that are sent at any of the following three labels:

- NEED_TO_KNOW LEGAL
- INTERNAL_USE_ONLY, and
- PUBLIC

The legal department’s printer set up as described above would reject jobs at any other sensitivity label. See Figure 14-2.

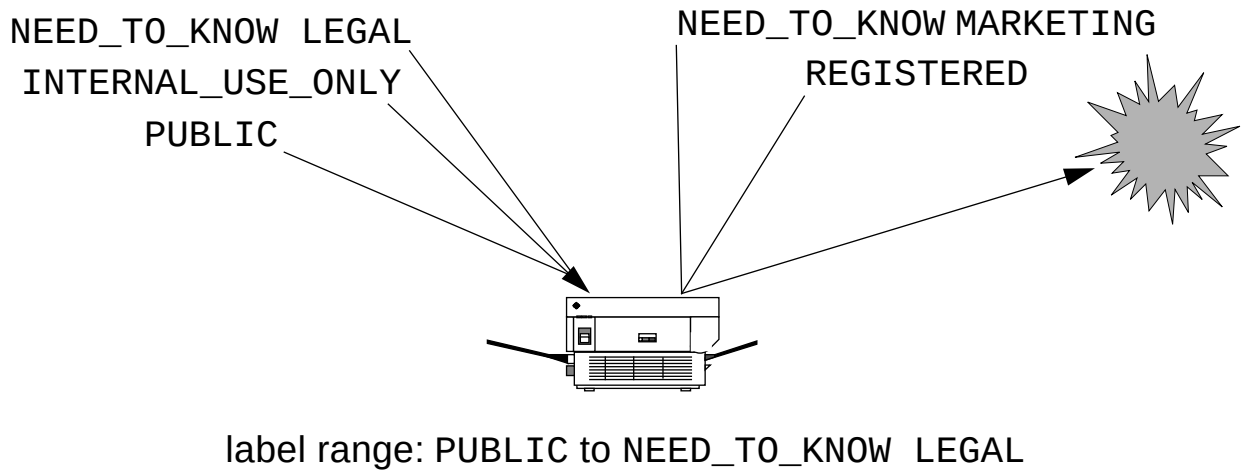



*Figure 14-2* Example of a Printer with a Restricted Label Range

The setting of a restricted label range for a printer is done by the administrator using the Device Allocation Manager, as described in:

| | |
|---|---|
| *To Configure a Restricted Label Range for a Printer Using the Device Manager* | *page 392* |

# 14

# Printing of Labels on Printer Output

A label is automatically printed at the top and bottom of each body page.

## Default: Information Label

By default, all jobs are printed with the *information label* of the job. Figure 14-3 shows a job whose information label is PUBLIC and sensitivity label is INTERNAL_USE_ONLY. The job printed with the PUBLIC information label at the top and bottom of every body page.

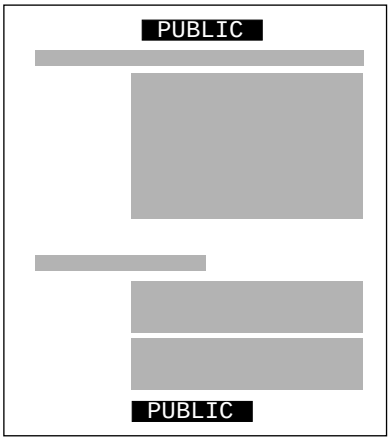



*Figure 14-3* Information Label Automatically Printed by Default on a Body Page

If information labels are disabled in the `system`(4) file (tsol_enable_IL=0), then a job's sensitivity label is printed instead of its information label. See *Trusted Solaris Label Administration* for more about the setting of kernel switches that enable and disable information labels by setting switches in the `system`(4) file. See also the `secconf`(2TSOL) man page.

## Alternate Body Page Labeling Options

The system administrator can decide whether to change the default so that one of the following is printed on body pages instead of the information label:

- The sensitivity label or
- Some other label or
- No label

Figure 14-4 shows email with the PUBLIC[INTERNAL_USE_ONLY} label sent to the printer at a site where information labels are disabled. Instead of the job's information label, the INTERNAL_USE_ONLY sensitivity label is printed.

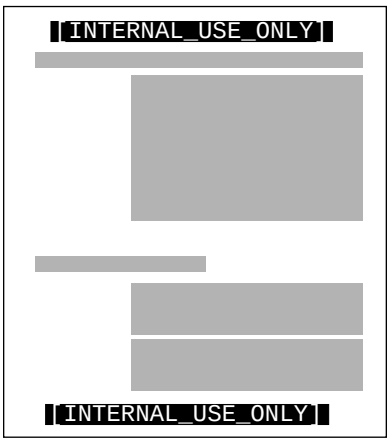



*Figure 14-4* Sensitivity Label Printed on Body Pages When Information Labels Are Disabled

## *Changing the Default Label on Body Pages*

The security administrator can change the default for the printing of labels on body pages in the following ways:

- Give users an authorization to allow them to print jobs without labels on the body pages

OR

- Change a configuration file on the print server so that:
  - Labels are not printed on body pages for any users
  - Some labels other than information labels are printed on body pages for all users

See these related procedures:

| | |
|---|---|
| *To Allow Some Users to Print Jobs Without Banners and Trailers* | *page 397* |
| *To Suppress the Printing of Page Labels on All Print Jobs* | *page 398* |

Also see "/usr/lib/lp/postscript/tsol_separator.ps" on page 372.

# 14

# *Labels, Job Numbers, and Handling Information Printed on Banner and Trailer Pages*

The banner and trailer pages that are automatically created for each print job are printed with company-specific handling guidelines. This section describes the types of labels and text that are printed by default on the banner and trailer pages.

Figure 14-5 on page 371 shows a banner page. Banner pages are printed before each job, unless they are suppressed as described under "To Allow Some Users to Print Jobs Without Banners and Trailers" on page 397.

Figure 14-6 on page 371 shows the differences on the trailer page. A thick black line appears instead of a gray frame, and the page type identifier changes from JOB START to JOB END.

## *Changing the Default Labels and Warnings on Print Jobs*

All the text and the labels and warnings that appear on print jobs are site-configurable. The labels that appear on the print job itself and the information shown in Figure 14-5, "Typical Print Job Banner Page," which appears on both banner and trailer pages, is configured in two places:

- `label_encodings(4TSOL)`
- `/usr/lib/lp/postscript/tsol_separator.ps`

### `label_encodings`*(4TSOL)*

The following portions of printer trailer and banner pages are configured in the `label_encodings(4TSOL)`:

- The "protect as classification,"
- The "access-related" words
- The handling instructions specified in the PRINTER BANNERS line(s)
- The handling instructions specified in the CHANNELS line(s)

See Chapter 3, "Specifying Labels and Handling Guidelines for Printer Output," in the *Trusted Solaris Label Administration* manual for how to configure these portions of the banner and trailer pages.

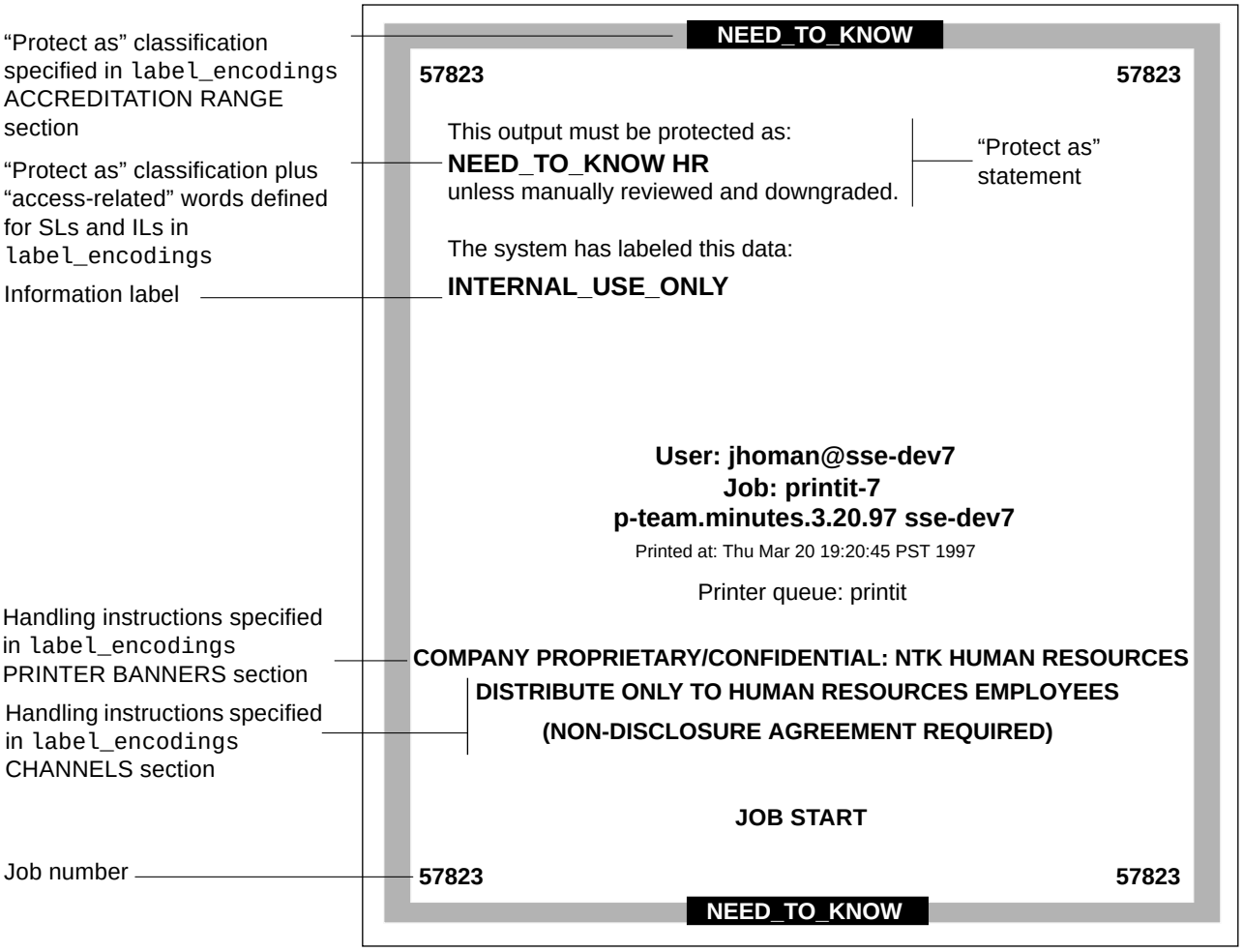



*Figure 14-5* Typical Print Job Banner Page

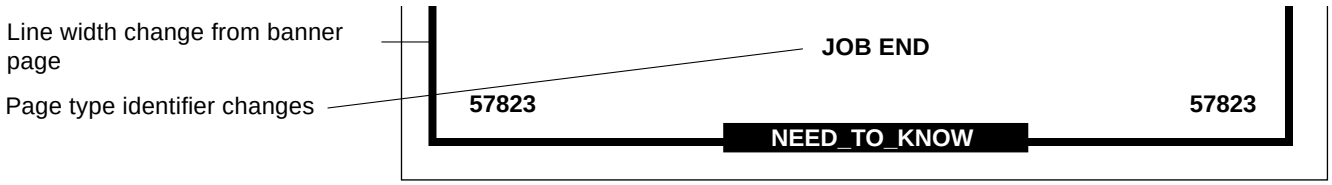



*Figure 14-6* Differences on a Trailer Page

# 14

## /usr/lib/lp/postscript/tsol_separator.ps

The security administrator may modify the `tsol_separator.ps` file in the `/usr/lib/lp/postscript` directory to do the following:

- Internationalize 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, or
- Change or omit any of the text or labels

The most-common substitution that sites choose to make is to specify that the sensitivity label prints instead of the information label at the top and bottom of banner pages. See:

*To Specify SLs to Print Instead of ILs on Body Pages* *page 396*

For how to do any other customizations or internationalization, see the comments in the `tsol_separator.ps` file. Table 14-1 shows the comments and modifiable fields in the `tsol_separator` file, with the rest of the programming code removed.

*Table 14-1* `tsol_separator.ps` Comments and Configurable Values *(1 of 7)*

```
%!
%ident"@(#)tsol_separator.ps5.297/05/28 SMI; TSOL 2.x"

%% Copyright (c) 1997 by Sun Microsystems, Inc.
%% All rights reserved.


%% This PostScript file is normally used as input to the lp.tsol_separator
%% program, which will prepend code to set the values of a number of
%% variables.  lp.tsol_separator is called by the printer interface script.

%% This PostScript file may be modified for local customizations or
%% internationalization.  Comments marked "INTERNATIONALIZE:" show
%% places where changes may be made for internationalization.  Comments
```

*Table 14-1* `tsol_separator.ps` Comments and Configurable Values *(2 of 7)*

```
%% marked “CUSTOMIZE:” show places where some typical customization
%% changes may be made.


%% The following comments describe variables set by lp.tsol_separator

%% These variables are from the print job information that can be
%% displayed with lpstat or lpq.
%%
%%   /Job_HeadLabel   The classification (from the sensitivity label) to
%%                    be displayed at the top and bottom of the banner
%%   /Job_Printer     Printer Name
%%   /Job_Host        Host job was submitted from
%%   /Job_User        User who submitted the job
%%   /Job_JobID       Job number
%%   /Job_Title       Job title
%%
%%
%% This variable is NO if an authorized user used the lp -o nobanner option
%% and the printer was set up to allow bannerless jobs.  Otherwise it is YES.
%%
%%   /Job_DoPageLabels Print page labels YES/NO.
%%
%%
%% These variables are generated from the system clock value.
%%
%%   /Job_Date        Date and time the job is being printed, in the
%%                    locale’s default format
```

*Table 14-1* `tsol_separator.ps` Comments and Configurable Values *(3 of 7)*

```
%%   /Job_Hash        A randomly generated identifying number for
%%                    matching up the banner and trailer pages of the job
%%
%%
%% The following variables are the job's Sensitivity Label and
%% Information Label as interpreted by the bcltobanner(3TSOL) library
%% routine.
%%
%%   /Job_HeadLabel   The classification (from the sensitivity label) to be
%%                    displayed at the top and bottom of the banner page.
%%   /Job_Protect     The sensitivity label to be displayed in the protect-as
%%                    field.
%%   /Job_Information The information label to be displayed in the "the system
%%                    has labeled this data as" field.  If ILs are turned off
%%                    in secconf, this variable will be an empty string.
%%   /Job_PageLabel   The information label to print at the top and bottom of
%%                    each page.  If ILs are turned off in secconf, this
%%                    variable will contain the sensitivity label instead.
%%   /Job_Caveats     The caveats from the information label.
%%   /Job_Channels    The channels from the information label.
%%
%%
%% The following variables are the job's Sensitivity Label and
%% Information Label as interpreted by the bsltos and biltos library
%% routines.
%%
%%   /Job_SL_Internal The sensitivity label in internal view format.
%%   /Job_SL_External The sensitively label in external view format.
```

*Table 14-1* `tsol_separator.ps` Comments and Configurable Values *(4 of 7)*

```
%%    /Job_IL_Internal The information label in internal view format.
%%    /Job_IL_External The information label in external view format.
        .
        .
        .
    %% INTERNATIONALIZE: Replace the text between
    %% parentheses with the appropriate text.
    %% CUSTOMIZE: The text between parentheses may be changed
    %% to use different wording, or changed to empty
    %% parentheses to eliminate the text.
    /ILabelText (The system has labeled this data:) def
    %% CUSTOMIZE: To not display the information label, change
    %% this line to: /ILabel () def
    /ILabel Job_Information def
        .
        .
        .
    %% CUSTOMIZE: To use a different string at the top and
    %% bottom of each page, change the following line.  For
    %% instance, to use the sensitivity label in external view
    %% format, change the line to: /PageLabel Job_SL_External def
    %% To eliminate page labels complete, change this line to
    %% set the page label to an empty string: /PageLabel () def
    /PageLabel Job_PageLabel def
        .
        .
        .
    %% INTERNATIONALIZE: Replace the text between
```

*Table 14-1* `tsol_separator.ps` Comments and Configurable Values *(5 of 7)*

```
    %% parentheses with the appropriate text.
    /Protect Job_Protect def
    /Protect_Text1 (This output must be protected as:) def
    /Protect_Text2
  (unless manually reviewed and downgraded.) def
        .
        .
        .
    %% CUSTOMIZE:  To not print the caveats, change
    %% this line to  /Caveats () def
    /Caveats Job_Caveats def
    %% CUSTOMIZE:  To not print the channels, change
    %% this line to  /Channels () def
    /Channels Job_Channels def

    %% CUSTOMIZE:  To not print the hash number, change
    %% this line to  /Hash () def
    /Hash Job_Hash def

    %% CUSTOMIZE:  To not print the head label, change
    %% this line to  /HeadLabel () def
    %% You may also substitute another string.  For example, to use
    %% the SL in internal view format: /HeadLabel Job_SL_Internal def
    /HeadLabel Job_HeadLabel def
     .
     .
     .
%% INTERNATIONALIZE: Replace the text between
```

*Table 14-1* `tsol_separator.ps` Comments and Configurable Values *(6 of 7)*

```
%% parentheses with the appropriate text.
(User:  ) User (@) Host append append append
    .
    .
    .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
(Job:  ) JobID append
    .
    .
    .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
(Printed at:  ) Date append
    .
    .
    .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
(Printer queue:  ) Printer append
    .
    .
    .

%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
{ TSOLJobInfo (JOB START) JobHashInfo}
    .
```

*Table 14-1* `tsol_separator.ps` Comments and Configurable Values *(7 of 7)*

```
.
.
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
{ TSOLJobInfo (JOB END) JobHashInfo}
.
.
.
%% End of tsol_separator.ps
```

## Supported Printers and Types of Output

PostScript printers are the only types of supported printers. Non-PostScript printers should function correctly but have no support for page labels or banner and trailer pages, so they do not meet Trusted Solaris printing requirements.

### Printing PostScript Files

By default, users may not print PostScript files. This restriction exists because a knowledgable PostScript programmer can create a PostScript file that modifies the labels on the printer output.

To allow an account to bypass this restriction, the security administrator role can assign the authorization called “print a PostScript file” to an account. The security administrator should do so only if the account doing the printing can be trusted not to spoof the labels on printer output and if allowing anyone to print PostScript files is within the site’s security policy. See “Authorizations to Bypass Printing Defaults” on page 383.

## Printing ASCII and Unsupported File Contents

ASCII is the only type of supported output. The only filter that is provided with the Trusted Solaris printing system converts text files to PostScript. Text files converted to PostScript by the installed filter program 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.

Because of the restriction on submitting user-generated PostScript, a site's administrator may choose to install additional filters to convert commonly used file types to PostScript that can also be trusted to have authentic labels and banner and trailer pages. See the Solaris 2.5.1 *System Administration Guide* for how filters are added.

## Sending Jobs to Printers Connected to Other Print Servers

A printer connected to a host (print server) that is not running Trusted Solaris can print jobs sent from a Trusted Solaris host. However, the jobs print without labels and without trailer pages and without security information on any banner pages, and therefore this type of printing does not meet Trusted Solaris requirements.

If consistent with a site's security policy, the security administrator can set up printing to a printer connected to a host (print server) that is not running Trusted Solaris by ensuring that both of the following are true:

- The security administrator has configured the print server host with a specific sensitivity label, and
- The print jobs are sent at the defined sensitivity label.

See Chapter 10, "Specifying Security Attributes in Trusted Network Databases and Setting Up Routing," for how the security administrator assigns a single sensitivity label to an unlabeled host.

## Permitting Publicly-readable Jobs to Be Printed by Default Without Labeled Pages

Certain users or groups of users, such as technical writers, need to produce publicly readable documents without labels printed on the top and bottom of the pages. If there is a printer connected to a Solaris print server available, the

security administrator can set up the users' environments so that the publicly-readable jobs go to the printer connected to the Solaris host while jobs at all other labels go to Trusted Solaris machines. The procedure requires understanding of how to set up user accounts as described in Chapter 3, "Managing User Accounts," and host network entries as described in Chapter 10, "Specifying Security Attributes in Trusted Network Databases and Setting Up Routing." See:

*To Set Up Automatic Printing of Publicly-readable Jobs Without Labels* *page 399*

# Configuring Printers

The system administrator configures printers using these administrative applications:

- The modified Solstice Printer Manager—to set up local and remote printers
- The Device Allocation Manager: Administration dialog box—to specify a restricted label range for any printer

Access the Solstice Printer Manager through the Application Manager Folder from the Solstice_Apps folder. See “To Access the Printer Manager” on page 384.

Access the Device Allocation Manager through the Trusted Desktop subpanel in the Front Panel or through the Allocate Device option on the Trusted Path menu. “To Configure a Restricted Label Range for a Printer Using the Device Manager” on page 392

See “To Install a Printer on a Print Server” on page 386 and “To Add Access to a Remote Printer” on page 394.

# 14

## *Modified Utilities and Man Pages*

The commands in the following list are modified to work within Trusted Solaris security policy. To use the administrative commands identified by the 1MTSOL man page suffix, the account must have the *administer printing* authorization. For details on these commands and the Trusted Solaris differences, see the `man(1TSOL)` pages.

- `accept(1MTSOL)`
- `cancel(1TSOL)`
- `disable(1TSOL)`
- `enable(1TSOL)`
- `lp(1TSOL)`
- `lpadmin(1MTSOL)`
- `lpc(1BTSOL)`
- `lpmove(1MTSOL)`
- `lpq(1BTSOL)`
- `lpr(1BTSOL)`
- `lprm(1BTSOL)`
- `lpsched(1MTSOL)`
- `lpshut(1MTSOL)`
- `lpstat(1TSOL)`
- `lpsystem(1MTSOL)`
- `lptest(1BTSOL)`
- `reject(1MTSOL)`

## *Authorizations to Bypass Printing Defaults*

The security administrator may give to some users an authorization to bypass some of the default features of Trusted Solaris printing.

- Print without banner and trailer pages
- Print a PostScript file
- Print without labels

A fourth authorization is needed to administer printing:

- Administer printing

Table 14-2 defines the authorizations that are related to printing. In order for a user or role to be able to do the action described under the Purpose heading, the authorization must be in one of the profiles assigned to the user or role.

*Table 14-2* Authorizations Related to Printing

| Name | Purpose |
|---|---|
| administer printing | Permits the user or role to administer printing using administrative utilities listed in “Modified Utilities and Man Pages” on page 382 to start and stop printing daemons, list and cancel other users' print jobs, and so forth. (Assigned by default to the system administrator (admin) role.) |
| print without banners | Permits the user or role to submit a print request using the **`lp -o nobanners`** option, to suppress the printing of banner and trailer pages. NOTE: For this to option to work, the Always Print Banners check box on the Printer Manager entry for the printer *must no*t be checked. |
| print a PostScript file | Allows a user to print a PostScript file. |
| print without labels | Allows a user to submit a print request that specifies (by means of the **`lp -o nolabels`** option) that labels will not be printed on the top and bottom of the print job’s body pages. |

# 14

# Printing-related Procedures

## ▼ To Access the Printer Manager

1. **Assume the security administrator role, and go to an ADMIN_LOW workspace.**
2. **Click the Application Manager icon in the Front Panel to open it.**
   See Figure 14-9.

*Figure 14-7* Application Manager Icon

3. **Double-click the Solstice_Apps icon.**
   See Figure 14-8.

*Figure 14-8* Solstice_Apps Icon

4. **Double-click the icon for the Printer Manager.**
   See Figure 14-9.

*Figure 14-9* Printer Manager Icon Selected in the Solstice_Apps Folder

The Printer Manager: Load dialog box displays with None as the only Naming Service option, as shown in Figure 14-10 on page 385.

*Figure 14-10* Printer Manager: Load Dialog Box With None as the Only Naming Service Option

**5. Choose None for the naming service.**

The Printer Manager displays, as shown in Figure 14-11 on page 385.

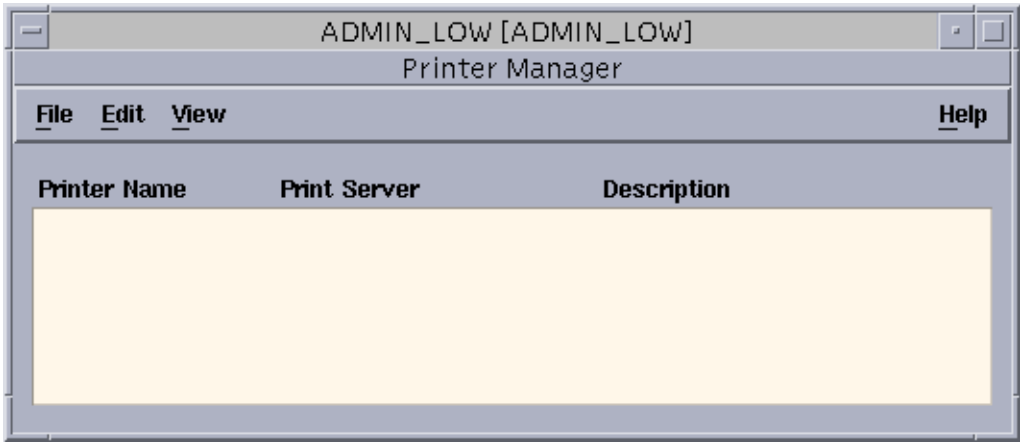



*Figure 14-11* Printer Manager

Go to “To Install a Printer on a Print Server” on page 386 or “To Add Access to a Remote Printer” on page 394.

## ▼ To Install a Printer on a Print Server

1. **Connect the printer to a serial or parallel port on a print serve using the appropriate cable, as described in the printer's installation manual.**
   The printer in the screen examples is connected to the serial port `/dev/term/b`.

2. **Click the Application Manager icon in the Front Panel to open it.**
   See Figure 14-12.

*Figure 14-12* Application Manager Icon

3. **Double-click the Solstice_Apps icon.**
   See Figure 14-13.

*Figure 14-13* Solstice_Apps Icon

4. **If the printer is connected to a serial port, make sure the correct baud rate is set.**

   **a. Double-click the icon for the Serial Manager.**
   See Figure 14-14.

*Figure 14-14* Serial Manager Icon

**b. On the Serial Port Manager dialog box, double-click the entry for the port to which the printer is connected.**
The Serial Port Manager: Modify dialog box displays. See Figure 14-15 on page 387.

**c. On the Serial Port Manager: Modify dialog box, verify that the baud rate specified for the port is set to 9600.**

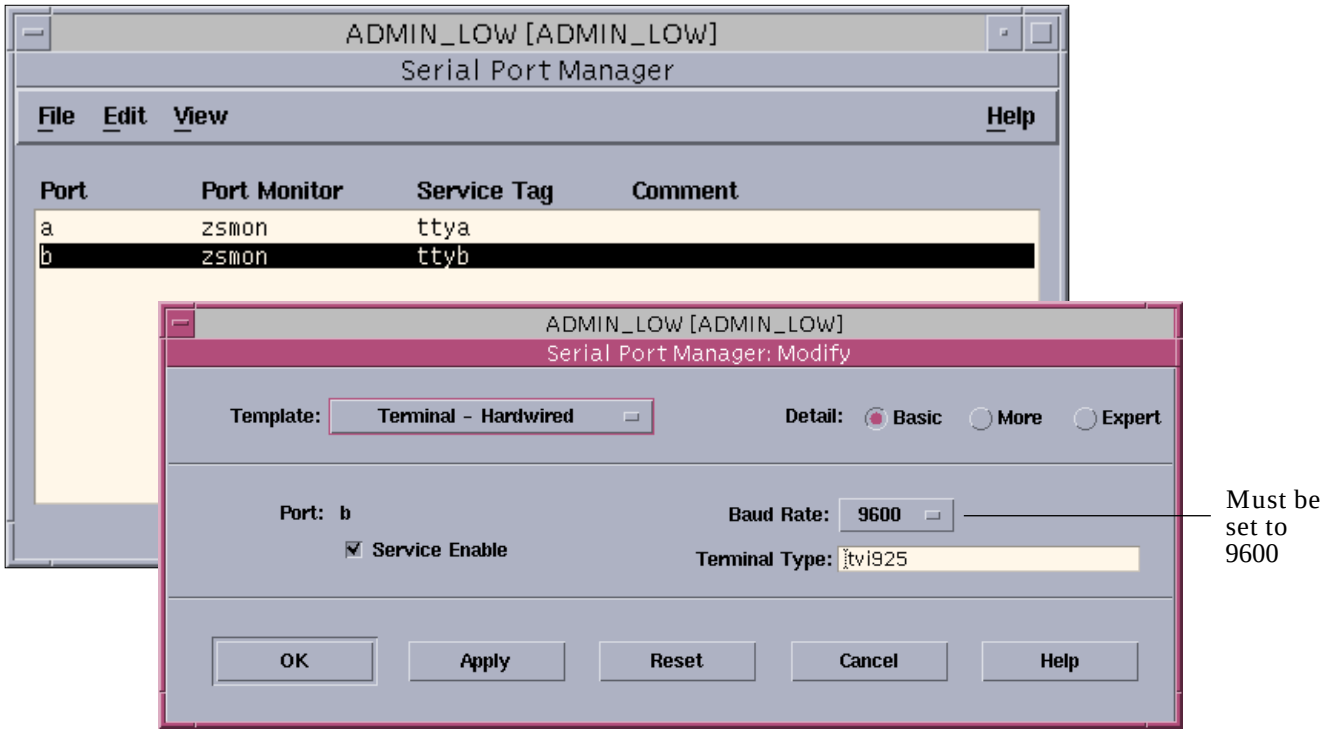



*Figure 14-15* Serial Port Manager and Serial Port Manager: Modify Dialog Boxes

**5. Access the Printer Manager from the Solstice_Apps folder.**
See “To Access the Printer Manager” on page 384, if needed.

6. **Choose Install Printer from the Edit menu, as shown in Figure 14-16**

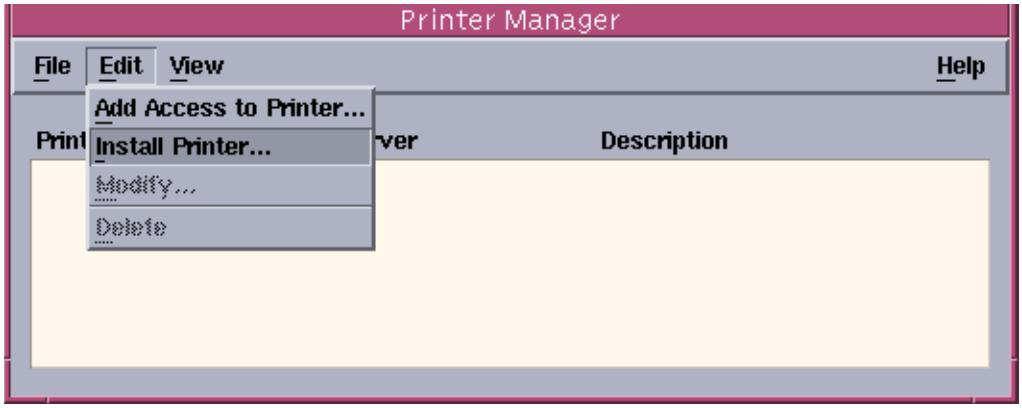



*Figure 14-16* Printer Manager: Selecting Install Printer from the Edit Menu

The Printer Manager: Install Printer dialog box displays as shown in Figure 14-17 on page 389.

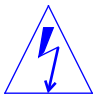
7. **Configure the printer.**

   a. **Supply a name for the printer.**

   b. **Supply a description if desired.**

   c. **From the Printer Port menu, select the name of the port where the printer is connected, or select Other and enter the name of an alternate port, if the printer is connected to a non-standard port.**

   d. **Leave Printer Type and File Contents set to PostScript.**

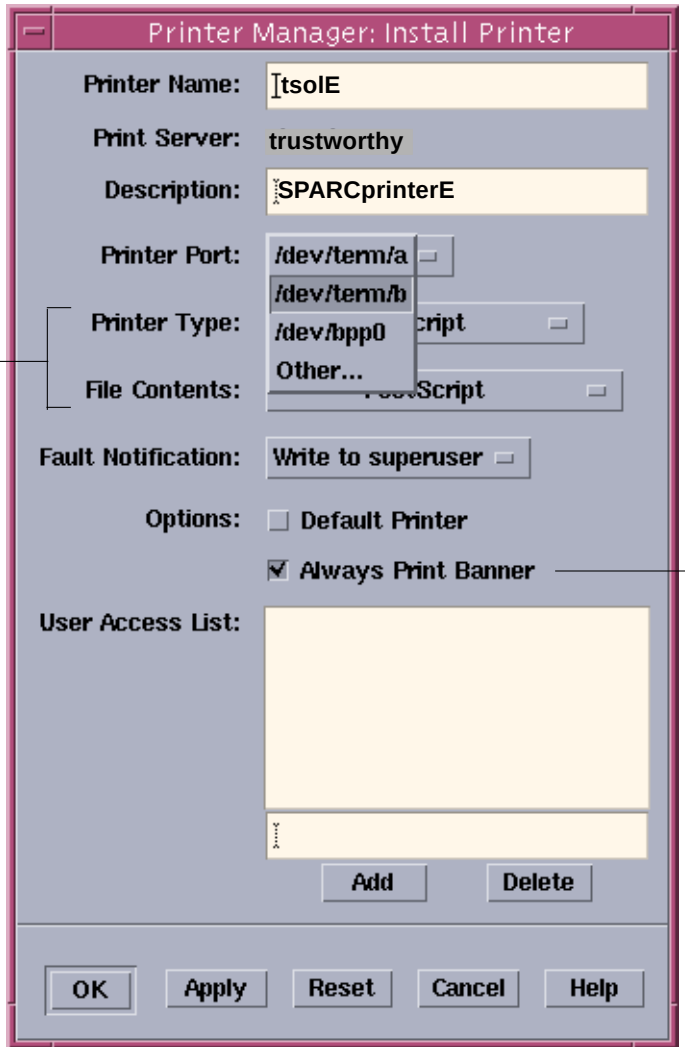


**Warning –** If you change the Printer Type and File Contents settings from the default value of PostScript, printing does not work.

   e. **Check the box next to Default Printer or not, as desired.**

   f. **Only if your site does not give any users the authorization to print without banner and trailer pages, check the box next to Always Print Banners.**

**Warning:** Do not change these defaults. Leave PostScript for the Printer Type and File Contents.




Leave this box unchecked if any users are authorized to print without banner and trailer pages

*Figure 14-17* Printer Manager: Install Printer Dialog Box

**g. If your sited gives any users the authorization to print without banner and trailer pages, do not check the box next to Always Print Banners.**

**h. Add users to the User Access List or not, as desired.**
Adding the username of any one user excludes all others.

**i. Click OK to save the changes and close the dialog box.**

8. **Go to the Application Manager > System_Admin folder, and use the Admin Editor action to open the `/etc/security/device_maps` file for editing.**

9. **Add an entry for the new printer by copying and modifying the entry for the `framebuffer`.**
   The framebuffer entry in `device_allocate`(4TSOL) can be most easily adapted, because the printer, like the framebuffer, is a non-allocatable device:.

```
framebuffer:fb:/dev/fb:
```

   Make the entry for the printer as shown here:

```
printername:lp:/dev/port_device_name_and_number:
```

   a. **Put the name of the printer in the first field.**

**Note –** Use the same name supplied for the printer in the Printer Manager.

   b. **Put the `lp` indicator in the second field.**

   c. **Put the pathname for the device special file for the port in the third field, ending with a colon.**
      The example shows an entry for a new printer named tsolE connected on serial port `/dev/ttya`.

```
tsolE:lp:/dev/ttya:
```

   d. **Save and quit the file.**

10. **Use the Admin Editor action to create an empty lock file in `/etc/security/dev`.**

   a. **Open `/etc/security/dev/`*printername*, with printername the same as the name of the new printer.**

   b. **Without editing the file, write it and quit it.**

c. **Change the user and group to bin, the mode to 000.**
The example shows a lock file being modified for a printer named tsolE. The printer lock file you create should have the same permissions, owner, group, and size shown in the long listing for the new empty lock file in the last line of the example.

```
$ chown bin tsolE
$ chgrp bin tsolE
$ chmod 000 tsolE
$ ls -l | grep tsolE
----------   1 bin      bin            0 Feb 24 11:58 tsolE
```

11. **Use the Admin Editor action to open the** `/etc/security/device_allocate` **file for editing.**

12. **Create an entry for the printer in** `device_allocate`(4TSOL) **by copying and modifying the entry for the** `framebuffer`**.**
The framebuffer entry in `device_allocate`(4TSOL) can be most easily adapted, because the printer, like the framebuffer, is a non-allocatable device:

```
framebuffer;fb;0x000...;0x7fff...;*;/bin/true
```

Make the entry for the printer as shown here.

```
printername;lp;0x000...;0x7fff...;*;/bin/true
```

a. **Put the name of the printer in the first field.**

b. **Put the** `lp` **name in the second field.**

c. **Leave the** `device_minimum` **and** `device_maximum` **label fields (fields 3 and 4) unmodified.**

d. **Make sure that the authorization field (field 5) has an asterisk (*), which indicates the device is not allocatable, and that the `device_clean` field (field 6) entry for printers is `/bin/true`.**
The example shows an entry for a new printer named tsolE.

```
tsolE;lp;0x000...;0x7fff...;*;/bin/true
```

e. **Save and quit the file.**

13. **To change the label range of the printer from ADMIN_LOW to ADMIN_HIGH, if desired, follow the steps under "To Configure a Restricted Label Range for a Printer Using the Device Manager" on page 392.**

## ▼ To Configure a Restricted Label Range for a Printer Using the Device Manager

1. **Assume the security administrator role and go to an ADMIN_LOW workspace on the print server.**
2. **Install the printer using the Printer Manager.**
See "To Install a Printer on a Print Server" on page 386.
3. **To change the label range from the default range of ADMIN_LOW to ADMIN_HIGH, use the Device Allocation Manager.**
   a. **As admin in an ADMIN_LOW workspace, 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 Trusted Desktop subpanel in the Front Panel.
   b. **Click the Device Administration button to display the Device Allocation: Administration dialog box.**
   c. **Highlight the name of the new printer in the Devices list.**
   d. **Click the Configure button to display the Device Allocation: Configuration dialog box.**
   **See Figure 14-12 on page 386.**

e. **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 sensitivity label.**

f. **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.**

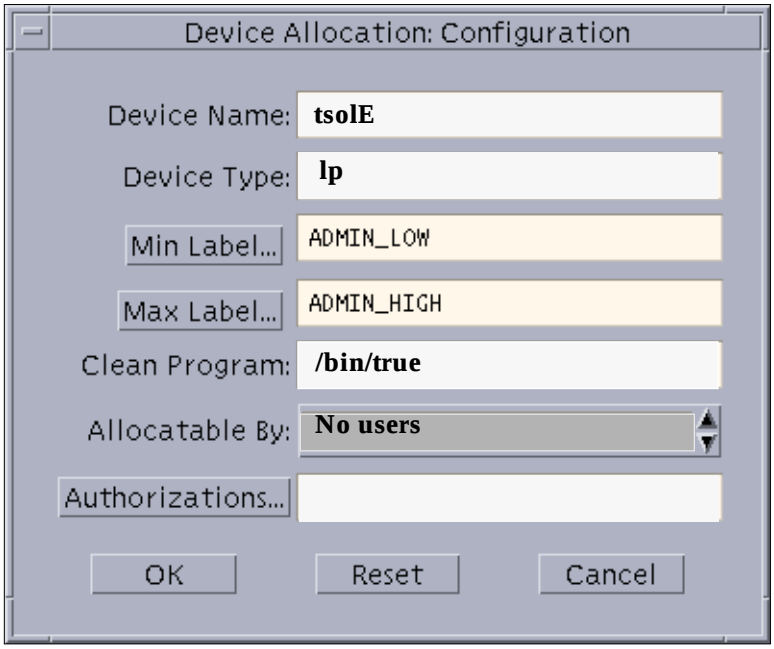



*Figure 14-18* Device Allocation: Configuration Dialog Box

## ▼ To Add Access to a Remote Printer

1. **Access the Printer Manager.**
   See “To Access the Printer Manager” on page 384, if needed.

2. **Choose Add Access to Printer from the Edit menu, as shown in Figure 14-19.**

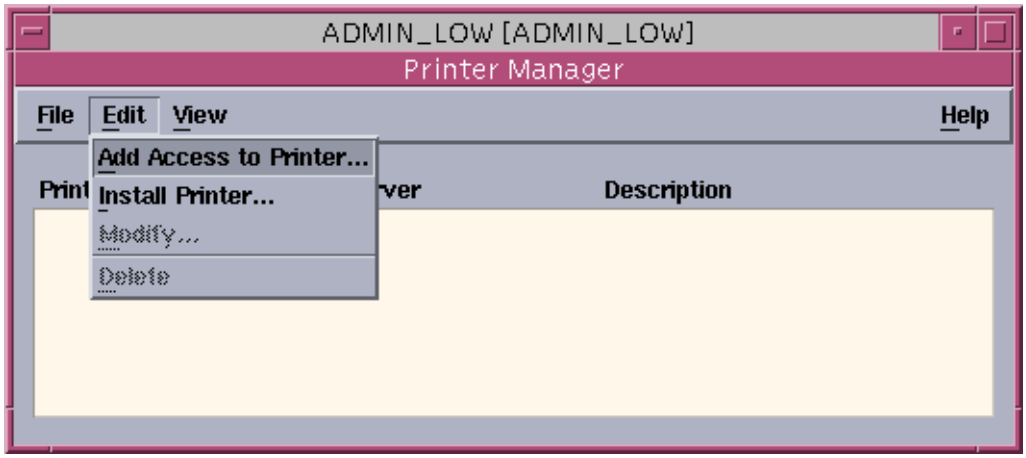



*Figure 14-19* Printer Manager: Selecting Add Access to Printer from the Edit Menu

The Printer Manager: Add Access to Printer dialog box displays, as shown in Figure 14-20 on page 395.

3. **Specify access to a remote printer from the local host.**
   a. **Type in the name of the remote printer in the Printer Name field.**
   b. **Type in the print server’s name in the Print Server field.**
   c. **OPTIONAL. Type a description in the Description field.**
   d. **OPTIONAL. Check the box next to Default Printer.**
   e. **Click OK to save the changes and close the dialog box.**

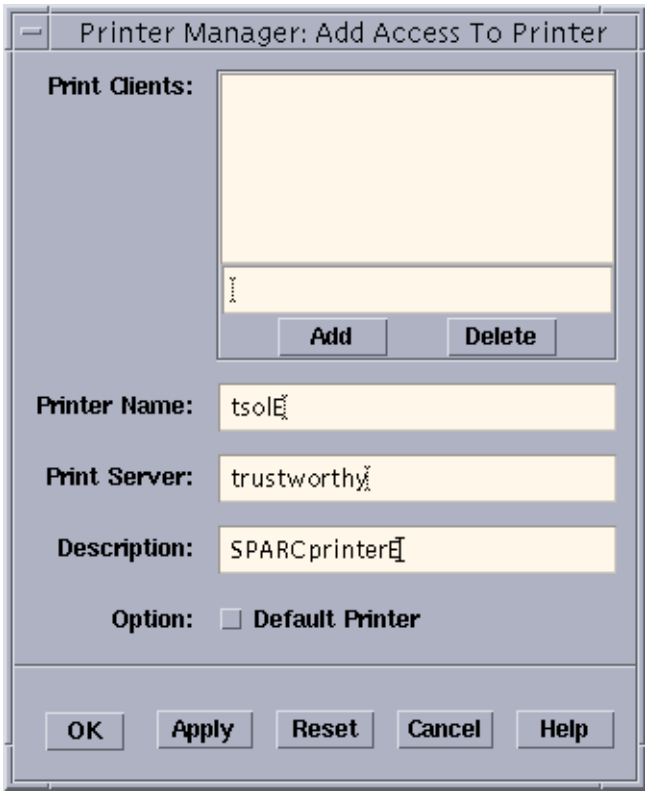



*Figure 14-20* Printer Manager: Add Access to Printer Dialog Box

4. **Specify access to a printer for a remote print client, if desired.**
   a. **In the field above the Add and Delete buttons, type in the name of a remote print client's name.**
   b. **Type in the name of the printer in the Printer Name field.**
   c. **Type in the print server's name in the Print Server field.**
   d. **OPTIONAL. Type a description in the Description field.**
   e. **OPTIONAL. Check the box next to Default Printer.**
   f. **Click the Add button to add the name of the print client to the list.**
      If you are done, go to Step g. To add more clients go back to Step a.
   g. **Click OK to save the changes and close the dialog box.**

### ▼ To Specify SLs to Print Instead of ILs on Body Pages

1. **Assume the security administrator role, and go to an ADMIN_LOW workspace**
2. **Go to the Application Manager > System_Admin folder and use the Admin Editor action to bring up the** `/usr/lib/lp/postscript/tsol_separator.ps` **file for editing.**
3. **Find the following line:**

```
/PageLabel Job_PageLabel def
```

4. **Change Job_PageLabel to whichever of the following settings you wish to use:**

```
%%   /Job_SL_Internal The sensitivity label in internal view format.

%%   /Job_SL_External The sensitivity label in external view format.

%%   /Job_IL_Internal The information label in internal view format.

%%   /Job_IL_External The information label in external view format.
```

The following line causes the external label view of the sensitivity label to be printed on body pages.

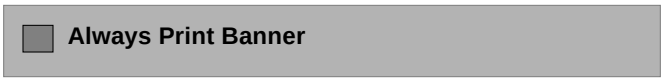
```
/PageLabel - def
```

5. **Save and close the file.**

## ▼ To Allow Some Users to Print Jobs Without Banners and Trailers

**Warning – If the Always Print Banner check box on the Printer Manager is checked, banner and trailer pages always print, even if the user has the print without banners authorization and uses the `-o` nobanners option to `lp`.**

1. **As security administrator, bring up the Printer Manager in an ADMIN_LOW shell.**
2. **Make sure that the Always Print Banner check box on the Printer Manager is *not* checked.**




3. **Make sure that the *print without banners* authorization is in one of the profiles assigned to each user or role that is allowed to print without banner and trailer pages.**
4. **Make sure that the user or role submits jobs using `lp` with the `-o nobanners` option.**

```
trustworthy% lp -o nobanners staff.mtg.notes
```

### ▼ 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 from the System_Admin folder to bring up the** `/usr/lib/lp/postscript/tsol_separator.ps` **file for editing.**
3. **Find the following lines:**

```
%% To eliminate page labels complete, change this line to
%% set the page label to an empty string: /PageLabel () def
          /PageLabel Job_PageLabel def
```

4. **Replace** `Job_PageLabel` **with an empty parentheses.**

```
%% To eliminate page labels complete, change this line to
%% set the page label to an empty string: /PageLabel () def
            /PageLabel () def
```

### ▼ To Allow Some Users to Print Jobs Without Page Labels

1. **Make sure that the *print without labels* 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.**
2. **Make sure that the user or role submits jobs using** `lp` **with the** `-o nolabels` **option.**

```
trustworthy% lp -o nolabels staff.mtg.notes
```

Doing this procedure allows the authorized user or role to print jobs without labels when working at any sensitivity label.

### ▼ To Set Up Automatic Printing of Publicly-readable Jobs Without Labels

1. **Make sure that the `tnrhdb/tnrhtp` entries for the Solaris print server specify the default sensitivity label as the site's public sensitivity label (such as PUBLIC or UNCLASSIFIED).**
2. **Do the following three steps for each user or role you want to allow to print all publicly-readable files without page labels.**
   a. **Make sure that the public label is in the personal sensitivity label range for each account.**
   b. **Instruct each user about how to appropriately define the PRINTER variable in the `.login` file in the user's publicly-labeled home directory SLD.**
      i. **Go to the publicly-labeled home directory SLD.**
      ii. **Open the `.login` file for editing.**
      iii. **Define the PRINTER variable to be the name of the printer connected to a Solaris print server.**
      iv. **Save and quit the file.**
   c. **Instruct each user about how to appropriately define the PRINTER variable in the `.login` file in all other SLDs to be a printer connected to a Trusted Solaris print server. Tell them to do the following for each SLD at which they work:**
      i. **Go to the home directory SLD.**
      ii. **Open the `.login` file for editing.**
      iii. **Define the PRINTER variable to be the name of a printer connected to a Trusted Solaris print server.**
      iv. **Save and quit the file.**
   d. **Have each affected account log out and log in again to put the changed printer definitions in effect.**
   e. **Have each affected account create and print jobs from the publicly-labeled SLD.**

# 14

# Managing Device Allocation and Setting Device Label Ranges 15

This chapter describes how to manage device allocation and how to set label ranges on two types of nonallocatable devices: hosts and printers. The chapter includes background and procedures in the following sections.


| | |
|---|---|
| *Needed Terms and Concepts* | *page 402* |
| *Security Risks Associated With Use of Devices* | *page 402* |
| *Clearing Objects Prior to Reuse* | *page 403* |
| *How Allocation/Deallocation Make Object Reuse Possible* | *page 403* |
| *How Manually-allocatable Devices are Allocated and Deallocated and Administered* | *page 403* |
| *How Devices With Removable Media Are Handled* | *page 406* |
| *Controlling the Location of Devices in the Trusted Solaris System* | *page 407* |
| *Device-Clean Scripts* | *page 407* |
| *Label Considerations* | *page 410* |
| *Proper Handling of Information Stored on Removable Media* | *page 411* |
| *Considerations When Importing and Exporting Information* | *page 411* |
| *Lock Files for Allocatable Devices* | *page 412* |
| *Allocate Error State* | *page 413* |
| *How Non-Allocatable Devices Are Managed* | *page 413* |
| *Default device_allocate File Contents and Settings* | *page 413* |

# 15


| | |
|---|---|
| *How Device Allocation Works* | *page 415* |
| *Setting Policy for a Device* | *page 419* |
| *Managing Device Allocation and Setting Device Label Ranges* | *page 419* |


This chapter provides the following device allocation procedures:


| | |
|---|---|
| *To Add a New Device Allocation Authorization* | *page 420* |
| *To Manage Devices Using the Device Allocation Manager* | *page 420* |
| *To Add a New Allocatable Device or Make an Existing Device Allocatable* | *page 423* |
| *To Add a New Non-Allocatable Device* | *page 430* |
| *To Change or Add a Device Clean Script* | *page 433* |


## Needed Terms and Concepts

This section reviews some terms and concepts introduced in the *Trusted Solaris User's Guide* and the *Trusted Solaris Administration Overview* and adds more detailed information needed to understand how to perform the tasks described in this chapter.

### Security Risks Associated With Use of Devices

As an example of the security risks associated with the use of various I/O devices, consider how tape devices are typically used. Because tape devices are typically accessible to all users on a distributed system, an unscrupulous user could covertly read information from another user's tape while the tape was in a commonly-accessible tape device. In general, the device-allocation mechanism reserves devices to access by one user at a time and makes sure the device is cleared of information before making it available to another user.

## *Clearing Objects Prior to Reuse*

*Object reuse* permits devices to be accessed by multiple users, by ensuring the following conditions

- Only the user (or process acting on behalf of the user) who has allocated a device can access information contained on the device while it is allocated.
- Sensitive information is cleared from the device at deallocation, before the device can be allocated again and reused by another user or process.

## *How Allocation/Deallocation Make Object Reuse Possible*

Some devices are automatically allocated and deallocated on behalf of the user. To meet the object reuse requirement, the Trusted Solaris allocation mechanism automatically clears information from some devices (such as computer memory and disks) between allocations. Other devices are user-allocatable.

## *How Manually-allocatable Devices are Allocated and Deallocated and Administered*

The `allocate(1MTSOL)` and `deallocate(1MTSOL)` commands and `device_clean(1MTSOL)` scripts are used in manual allocation, deallocation, and clearing of other devices that are not automatically allocated and deallocated. Devices are allocated, configured, and managed through the Device Allocation Manager. Figure 15-1 shows the Device Allocation Manger, which can be invoked only by an account that has the device allocation authorization. The Device Maintenance button on the bottom displays only for the administrative role.

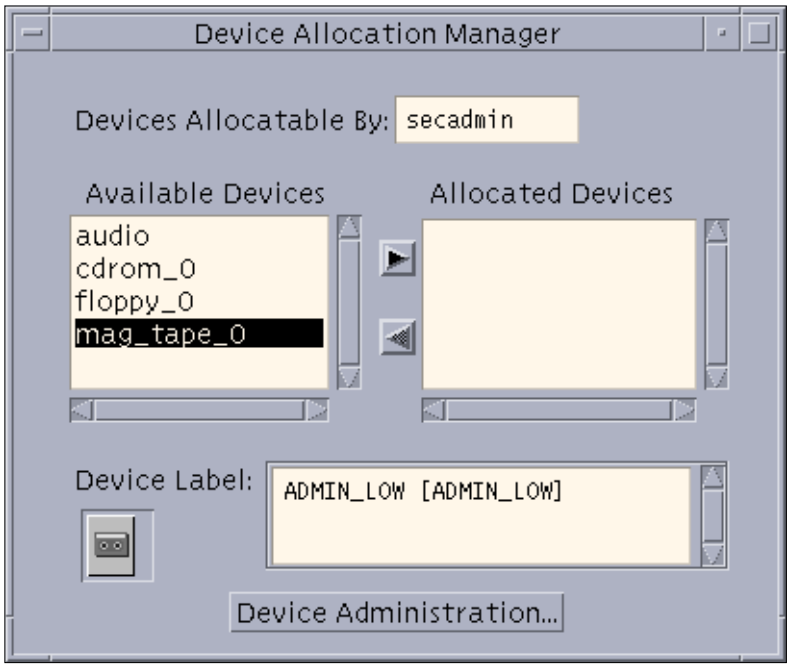



*Figure 15-1* Device Allocation Manager

Figure 15-2 on page 405 shows, left to right and top to bottom, the Device Allocation Manager, the Device Allocation: Administration dialog box, the Device Allocation: Configuration dialog box, and the Device Allocation: Authorizations dialog box.

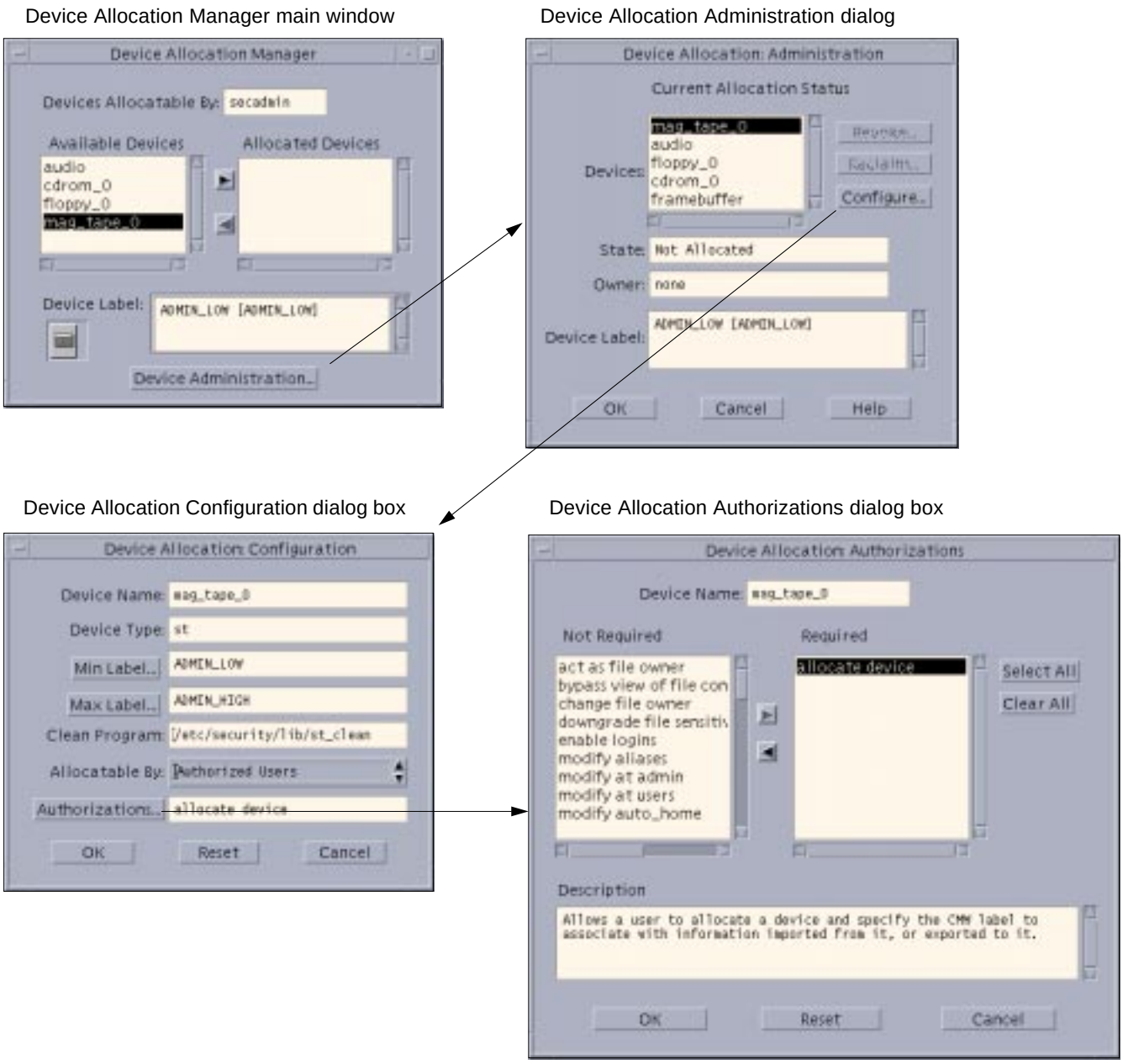



*Figure 15-2* Device Allocation, Management, Configuration and Authorizations Tools

As shown in Figure 15-2 on page 405, when the *Device Administration* button is clicked on the Device Allocation Manager, the *Device Allocation: Administration* dialog box displays. Clicking the *Revoke* Button forces deallocation of a selected device, clicking the *Reclaim* button restores a selected device from the allocate error state, and clicking the *Configure* button, also as shown, brings up the *Device Allocation: Configuration* dialog box. On the Device Allocation: Configuration dialog box, the *Device Name*, *Device Type*, and *Clean Program* fields are text entry fields. The *Min Label* and *Max Label* buttons bring up Label Builders, and the *Allocatable By* menu lets the administrator choose between making a device allocatable by *All Users*, *No Users*, or *Authorized Users*. When the *Authorizations* button is clicked, the *Device Allocation: Authorizations* dialog box displays, in which the administrator can choose to change the default *allocate device* authorization to another one in the list of available authorizations, or choose to add multiple authorizations to the *Required* list. For adding a new authorization, if needed, see "To Add a New Device Allocation Authorization" on page 420. After a new authorization is added, it appears in the Not Required list in the Device Allocation: Authorizations dialog box.

For details on using the Device Allocation Manager, see "To Manage Devices Using the Device Allocation Manager" on page 420.

## How Devices With Removable Media Are Handled

Some of the devices that are manually allocated and deallocated allow information to be imported to and exported from the Trusted Solaris system by means of removable media. Because the removable media used in devices such as tape and floppy drives should not be cleared, device clean scripts either eject the media or prompt the user to eject the media before freeing the device for other users.

For devices that store information on and retrieve it from removable media, object reuse requires that the media be removed from the device before it is allocated to another user. The media must also be properly labeled with a physical label that reflects the CMW label (INFORMATION LABEL [SENSITIVITY LABEL] combination) of the stored information. Device clean scripts executed by the Trusted Solaris `deallocate` command enforce this requirement by prompting the user to eject the media before completing deallocation and by providing the CMW label to use on the media's physical label. The `deallocate` command succeeds only after the media has been ejected.

## Controlling the Location of Devices in the Trusted Solaris System

Successful control of access to devices partly depends on the devices being found only in certain locations in the file system. In the Trusted Solaris system, all device special files (which are used to manipulate devices) should be under the root (/) file system in the `/dev` and the `/devices` directory. All other file systems should be mounted with the Trusted Solaris `nodev` option, which prevents users from being able to bypass the device allocation mechanism by using devices located outside of the root file system.

For more about using the nodevices option, see "To Specify Mount-time Security Attributes in the Mount Table" on page 340. See also the `mount(1MTSOL)` man page.

## Device-Clean Scripts

A device-clean script is run any time a device is deallocated (either by a user's deallocating it or by an administrator's action to forcibly deallocate it). If your site adds additional allocatable devices to the system, new scripts may need to be created. See the following descriptions of the existing device-clean scripts for ideas on how they work.

## Device-Clean Script for Tape Devices

Table 15-1 shows the three supported tape devices and the name of the device-clean script used for all.

*Table 15-1* Device-Clean Scripts for the Three Supported Tape Devices

| Tape Device Type | Device-Clean Script |
|---|---|
| SCSI 1/4 inch tape | `st_clean` |
| Xylogics 472 1/2 inch tape | `st_clean` |
| Archive 1/4 inch tape | `st_clean` |

The `st_clean` script uses the `rewoffl` option to `mt(1)` to do the device-cleanup. If the script is run during system boot, it queries the device to see if it is on line and has any storage medium in it. If necessary, the script prompts the user to eject the storage media, and then it displays the appropriate CMW 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, to force the security administrator to manually clean up the device before any user can allocate it again if a user has not completed deallocation.

## Device-Clean Scripts for Floppy Disks and CD-ROM

Table 15-2 shows the device-clean scripts for the floppy disk drives and CD-ROM devices.

*Table 15-2* Device-Clean Scripts for Floppy and CD-ROM

| Disk Device Type | Device-Clean Script |
|---|---|
| floppy | `fd_clean` |
| CD-ROM | `sr_clean` |

These scripts use the `eject(1)` command to eject the medium from the drive. If these scripts are run during boot time, the appropriate CMW label for the medium displays on the console; at other than boot time, the scripts prompt the user to affix to the medium a physical label with the appropriate CMW label. If `eject(1)` fails, the device is placed in the allocate error state

## *Device-Clean Script for Audio*

The audio device is cleaned up using the `audio_clean(1)` program.

*Table 15-3* Device-Clean Script for Audiotool

| Disk Device Type | Device-Clean Script |
|---|---|
| audiotool | `audio_clean` |

This program performs an AUDIO_DRAIN `ioctl` to flush the device, and then an AUDIO_SETINFO `ioctl` to reset the device configuration to 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 medium, it does not require an external physical label, and therefore the CMW label is not displayed by the `audio_clean` script.

## *Writing New Device-Clean Scripts*

Devices that a site can make allocatable are modems, terminals, graphics tablets, and the like. A new device-clean scripts would need to be created for each of these devices, and the script would need to fulfill object-reuse requirements for the type of device. Device_clean scripts should also be created for any added tape devices, except for a Xylogics or an Archive tape drive, which can use the default `st_clean` script (`/etc/security/lib/ST_CLEAN`). See the `device_clean(1MTSOL)` man page.

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` command passes 4 parameters to the device-clean scripts as shown here:

```
st_clean -[I|F|S] -[A|D] device_name cmw_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` or `deallocate`.

*device_name* is a string with the name of the device

*cmw_label* is a hexadecimal representation of the CMW label

## Label Considerations

For normal users, information can only be exported or imported at a single sensitivity label, which is the sensitivity label at which the user allocates a device.

### Restricted Label Range

Label ranges may be set for certain types of nonallocatable devices: hosts and printers. The default label range set for the frame buffer may be modified so that access to a Trusted Solaris host is restricted; and when a host has a printer attached, the label range for that printer may also be modified to limit the label range of jobs that it will print.

A restricted label range may be used to reflect the security of the physical location of a device or to limit access to only those users with certain clearances.

### *MAC Rules for Device Access*

MAC rules control access to a device. The label range of the device is checked against the sensitivity label of the user's process. The sensitivity label at which the device is accessed must be within the device's label range. If the user's process sensitivity label is not within the label range of the device, allocation of the device is denied.

Allocation of a device assigns a sensitivity label to all the device special files associated with the device. The sensitivity label assigned to the device special files is determined by the process of the user allocating the device.

## *Proper Handling of Information Stored on Removable Media*

The security administrator must instruct the site's users about appropriate physical and procedural controls for tapes, CD-ROMs, and floppy disks, because removable media must be labeled properly, and access to information contained on the removable media must be controlled.

After information is exported to the tape or floppy, the user must eject the media from the device and label it with the sensitivity label and information label (CMW label) provided by the Trusted Solaris system at the time of deallocation. The medium must be stored and access to it must be controlled in such a way to protect it from unauthorized access.

## *Considerations When Importing and Exporting Information*

### *Using* `tar` *(1TSOL) With Device Allocation*

The only normal user command modified to store and retrieve Trusted Solaris security attributes is `tar(1TSOL)`, which stores security attributes for each file when the T option is used with the `c` and `r` options and retrieves security attributes when `T` is used with the `x` and `t` options.

---

**Note –** If a `tar` command fails when it attempts to access a device, the device may not have been allocated.

---

### *To Export Information*

A media device (for a tape or floppy) must be allocated at the sensitivity label of the information that is to be stored on the device.

### *To Import Information*

When a device that contains a tape or floppy used for storage of information is allocated for reading, the user must allocate the tape drive at the sensitivity label shown on the physical label on the tape.

### *To Import Information Whose Information Label Is Lower Than Its Sensitivity Label*

If the information label specified on a floppy or tape's physical label is lower than the sensitivity label, a user with the downgrade information authorization may decide to downgrade the sensitivity label of the information to match the information label. The device would be then allocated at a sensitivity label equal to the information label specified on the physical label, which would result in the information being imported at a sensitivity label equal to the tape or floppy's information label instead of at its sensitivity label.

## *Lock Files for Allocatable Devices*

Each allocatable device needs a *lock file*, which is a zero-length file created in `/etc/security/dev`. Lock files are sometimes referred to as DAC files because they must have a certain set of DAC permissions, owner, and group in order to be allocatable. If no lock file exists for an allocatable device, or if the characteristics are not correct, the device cannot be allocated, and no one can access the device.

*Table 15-4* Required Lock File Characteristics for Allocatable and Allocated Devices

| | **DAC permissions (mode)** | **Owner** | **Group** |
|---|---|---|---|
| **Allocatable** | 000 | bin | bin |
| **Allocated** | 600 | user | user's group |

## *Allocate Error State*

As shown in Table 15-5, an allocatable device is in the *allocate error state* if it is owned by user `allocate` and group `wheel` with a device special file mode of 0100 and sensitivity label of ADMIN_HIGH. Device clean scripts put a device into the allocate error state until an administrative role uses the Reclaim button on the Device Allocation: Maintenance dialog box (as shown in "Device Allocation: Administration Dialog Box" on page 422).

*Table 15-5* Lock File Settings for Device in the Allocate Error State

| | DAC permissions (mode) | Owner | Group | sensitivity label |
|---|---|---|---|---|
| **Error State** | 100 | bin | bin | ADMIN_HIGH |

## *How Non-Allocatable Devices Are Managed*

The frame buffer is the only non-allocatable device with an entry in the default `label_encodings` file. A site that wishes to restrict access to an individual workstation or server should restrict the label range on the framebuffer. Entries for printers can also be added, as described in "To Configure a Restricted Label Range for a Printer Using the Device Manager" on page 392, when a print server is being configured. The frame buffer and any added printers should be configured by default with `ADMIN_LOW to ADMIN_HIGH` label range and with an asterisk(`*`) in the authorization field to indicate that the device is not allocatable. Currently, these are the only two device types whose label range may be restricted in the `device_allocate` file. Changes to the label range of any other device has no effect.

## *Default* `device_allocate` *File Contents and Settings*

Each allocatable device has the following characteristics configured by default:

- Sensitivity label range: ADMIN_LOW ADMIN_HIGH
- A device-clean script
- A lock file created in `/etc/security/dev` with mode 000, owner allocate, group wheel, and a CMW label of ADMIN_LOW[ADMIN_LOW].

The frame buffer and printer entries are configured by default with `ADMIN_LOW;ADMIN_HIGH` label ranges and with an asterisk(`*`) in the authorization field to indicate that the device is not allocatable.

Figure 15-3 shows the contents of the default `device_allocate(4TSOL)` file while Figure 15-3 gives the default settings in tabular form.

```
mag_tape_0;st;0x0000000000000000000000000000000000000000000000000000000000000000;0x7fffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffff;allocate device;/etc/security/lib/st_clean
audio;audio;0x0000000000000000000000000000000000000000000000000000000000000000;0x7fffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffff;allocate device;/etc/security/lib/audio_clean_wr
apper
cdrom_0;sr;0x0000000000000000000000000000000000000000000000000000000000000000;0x7fffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffff;allocate device;/etc/security/lib/sr_clean
framebuffer;fb;0x0000000000000000000000000000000000000000000000000000000000000000;0x7fffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffff;*;/bin/true
```

*Figure 15-3* The Default device_allocate File

*Table 15-6* Default Settings the `device_allocate` File

| Name | Type | Description | Label Range | Authorization Field | Device-Clean Script |
|---|---|---|---|---|---|
| `mag_tape_0,` | st | SCSI device | `ADMIN_LOW; ADMIN_HIGH` | `ALLOC` | `st_clean` |
| `cdrom_0` | sr | CD-ROM | `ADMIN_LOW; ADMIN_HIGH` | `ALLOC` | `sr_clean` |
| `audio` | audio | audio chip | `ADMIN_LOW; ADMIN_HIGH` | `ALLOC` | `audio_clean` |
| `framebuffer` | fb | frame buffer (sets the host's label range) | `ADMIN_LOW; ADMIN_HIGH` | * | none |

## How Device Allocation Works

1. When a device is allocated, the `allocate(1MTSOL)` command checks for the presence of a lock file under the device name in the `/etc/security/dev` directory.

   Figure 15-4 shows a long listing of the lock file for the `mag_tape_`*N* device in `/etc/security/dev`. As required for the `allocate` command to succeed, the lock file is owned by `bin` with group `bin`, and it has a mode of `000`.

```
falloon%   cd /etc/security/dev
falloon% ls -l
total 0
----------   1 bin      bin            0 Feb 24 21:58 mag_tape_0
----------   1 bin      bin            0 Feb 24 21:59 mag_tape_1
```

*Figure 15-4* Listing for an Allocatable Device's Lock File

2. If the lock file exists with the proper owner, group and permissions, the `allocate` command checks for the presence of the necessary device special files associated with the device in the `/dev` directory.

   Figure 15-5 on page 416 shows the use of `ls -lg` in the `/dev` directory. The listing shows that the device special files for the tape device are set up properly, with owner `allocate`, group `wheel`, and mode `000`.

```
untouchable% ls -lg /dev
c--------- 1 bin     bin          18,  4 May 12 13:11 nrst5
c--------- 1 bin     bin          18, 20 May 12 13:11 nrst13
c--------- 1 bin     bin          18, 28 May 12 13:11 nrst21
c--------- 1 bin     bin          18, 12 May 12 13:11 nrst29 .
                           .
                           .
c--------- 1 bin     bin          18, 0 May 12 13:11 rst5
c--------- 1 bin     bin          18, 16 May 12 13:11 rst13
c--------- 1 bin     bin           18, 24 May 12 13:11 rst21
c--------- 1 bin     bin          18, 8 May 12 13:11 rst29

untouchable% ls -lg /dev/rmt/
c--------- 1 bin     bin          18, 0 May 12 13:11 0
c--------- 1 bin     bin          18, 16 May 12 13:11 0n
c--------- 1 bin     bin           18, 24 May 12 13:11 0b
c--------- 1 bin     bin          18, 8 May 12 13:11 0l
c--------- 1 bin     bin          18, 0 May 12 13:11 0m
c--------- 1 bin     bin          18, 16 May 12 13:11 0h
c--------- 1 bin     bin           18, 24 May 12 13:11 0c
c--------- 1 bin     bin          18, 8 May 12 13:11 0u
c--------- 1 bin     bin          18, 8 May 12 13:11 0ln
c--------- 1 bin     bin          18, 0 May 12 13:11 0mn
c--------- 1 bin     bin          18, 16 May 12 13:11 0hn
c--------- 1 bin     bin           18, 24 May 12 13:11 0cn
c--------- 1 bin     bin          18, 8 May 12 13:11 0un
c--------- 1 bin     bin           18, 24 May 12 13:11 0ub
c--------- 1 bin     bin          18, 8 May 12 13:11 0bn
c--------- 1 bin     bin          18, 0 May 12 13:11 0ubn
```

*Figure 15-5* Listing for Device Special Files for an Allocatable Tape Device

3. If the lock file and the correct device special files do not exist or if they do not have the correct user, group, and permissions, then the device cannot be allocated by the current user, and the attempted allocation fails.

4. If the previous checks succeed, the `allocate` command then checks for an entry for the device in the `device_allocate` file, and checks whether the device's entry indicates that the device is allocatable.

Figure 15-6 shows the entry for mag_tape_0 in the `device_allocate` file with the string “allocate device” in the allocate field, which indicates that the device is allocatable. (See the `device_allocate(4TSOL)` man page.)

```
mag_tape_0;st;0x0000000000000000000000000000000000000000000000000000000000000000;0x7fffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffff;allocate device;/etc/security/lib/st_clean
```

*Figure 15-6* Device Allocate File Entry for mag_tape_0

5. If all the previously listed conditions are met, the device is allocated to the user.

   Allocation changes the ownership and permissions of the device special files associated with the device in the `/dev` directory. Figure 15-7 shows the device special files associated with a tape device in the `device_maps(4TSOL)` file.

```
mag_tape_0:\
        st:\
        /dev/rst5 /dev/nrst5 /dev/rst13 /dev/nrst13 /dev/rst21
/dev/nrst21 /dev/rst29 /dev/nrst29 /dev/rmt/0 /dev/rmt/0n
/dev/rmt/0b /dev/rmt/0l /dev/rmt/0m /dev/rmt/0h /dev/rmt/0c
/dev/rmt/0u /dev/rmt/0ln /dev/rmt/0mn /dev/rmt/0hn /dev/rmt/0cn
/dev/rmt/0un /dev/rmt/0ub /dev/rmt/0bn /dev/rmt/0ubn:\
```

*Figure 15-7* Device Special Files Associated With the Tape Device

Figure 15-8 shows the changes to the permissions and ownership of the associated device special files when user vanessa allocates the tape device. As shown in the partial listing of the associated device special files, the mode is changed to `600`, the owner is changed to vanessa, and the group is changed to staff.

```
untouchable% ls -lag /dev
crw------- 1 vanessa staff        18,  4 May 12 13:11 nrst5
crw------- 1 vanessa staff        18, 20 May 12 13:11 nrst13
crw------- 1 vanessa staff        18, 28 May 12 13:11 nrst21
crw------- 1 vanessa staff        18, 12 May 12 13:11 nrst29 .
                            .
                            .
crw------- 1 vanessa staff        18, 0 May 12 13:11 rst5
crw------- 1 vanessa staff        18, 16 May 12 13:11 rst13
crw------- 1 vanessa staff        18, 24 May 12 13:11 rst21
crw------- 1 vanessa staff        18, 8 May 12 13:11 rst29

untouchable% ls -lg /dev/rmt/
crw------- 1 vanessa staff        18, 0 May 12 13:11 0
crw------- 1 vanessa staff        18, 16 May 12 13:11 0n
crw------- 1 vanessa staff        18, 24 May 12 13:11 0b
crw------- 1 vanessa staff        18, 8 May 12 13:11 0l
crw------- 1 vanessa staff        18, 0 May 12 13:11 0m
crw------- 1 vanessa staff        18, 16 May 12 13:11 0h
crw------- 1 vanessa staff        18, 24 May 12 13:11 0c
crw------- 1 vanessa staff        18, 8 May 12 13:11 0u
crw------- 1 vanessa staff        18, 8 May 12 13:11 0ln
crw------- 1 vanessa staff        18, 0 May 12 13:11 0mn
crw------- 1 vanessa staff        18, 16 May 12 13:11 0hn
crw------- 1 vanessa staff        18, 24 May 12 13:11 0cn
crw------- 1 vanessa staff        18, 8 May 12 13:11 0un
crw------- 1 vanessa staff        18, 24 May 12 13:11 0ub
crw------- 1 vanessa staff        18, 8 May 12 13:11 0bn
crw------- 1 vanessa staff        18, 0 May 12 13:11 0ubn
```

*Figure 15-8* Checking the Existence, Ownership, Group, and Mode of Device Special Files for `mag_tape_0`

c. Allocating a command changes the ownership, group, and mode of the lock file associated with the device in the `/etc/security/dev` directory.

Figure 15-9 shows that the owner of /etc/security/dev/st0 is changed to vanessa and the mode is changed to 600 (read and write for the owner.

```
untouchable% ls -lg /etc/security/dev/st0
-rw------- 1 vanessa staff       0 Dec 6 15:21 /etc/security/dev/mag_tape_0
```

*Figure 15-9* Changing Ownership, Group, and Mode of st0's Lock File Upon Allocation

### *Setting Policy for a Device*

The device mechanism has a number of access control policies already defined. These defined policies may be assigned to new devices using the devpolicy(1MTSOL) command, which uses information from the device_policy(4TSOL) file.

## *Managing Device Allocation and Setting Device Label Ranges*

After installation of the software and during configuration, the security administrator initially defines which devices should be allocatable and what the label range should be on the appropriate devices. The security administrator role either accepts or changes the default configuration for devices and their defined characteristics. After the system is up and running, if a new device is added, the security administrator must decide whether to make the new device allocatable.

By using the Device Maintenance dialog box accessed from the Device Allocation tool, the security administrator role can make the following changes to the defaults. (See the procedures starting on page 420.)

- Make a new device or existing device allocatable
- Make an allocatable device available to all users, no users, or authorized users
- Restrict the label range on a (non-allocatable) host or printer.

# 15

# *Device Management Procedures*

## ▼ To Add a New Device Allocation Authorization

1. **Read "Extending Extendable Security Mechanisms" on page 31 of Chapter 2, "Miscellaneous Tasks and Procedures."**
2. **Do the procedure under "To Add An Authorization" on page 36 in Chapter 2.**

## ▼ To Manage Devices Using the Device Allocation Manager

1. **The security administrator role should make the following decisions before starting:**
   a. **Find out which devices are listed in the `device_allocate` file, what their default characteristics are, and which devices can be made allocatable.**
   b. **Define which devices, if any, should be allocatable at your site.**
   c. **Decide whether to make additional devices allocatable.**
   d. **Decide which normal users, if any, should be allowed to allocate devices.**
   e. **Decide whether to accept or change the default label range settings on the listed non-allocatable devices.**
2. **Assume the security role in an ADMIN_LOW workspace, and 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 Trusted Desktop subpanel in the Front Panel. The Device Allocation Manager displays, as shown in Figure 15-10 on page 421.

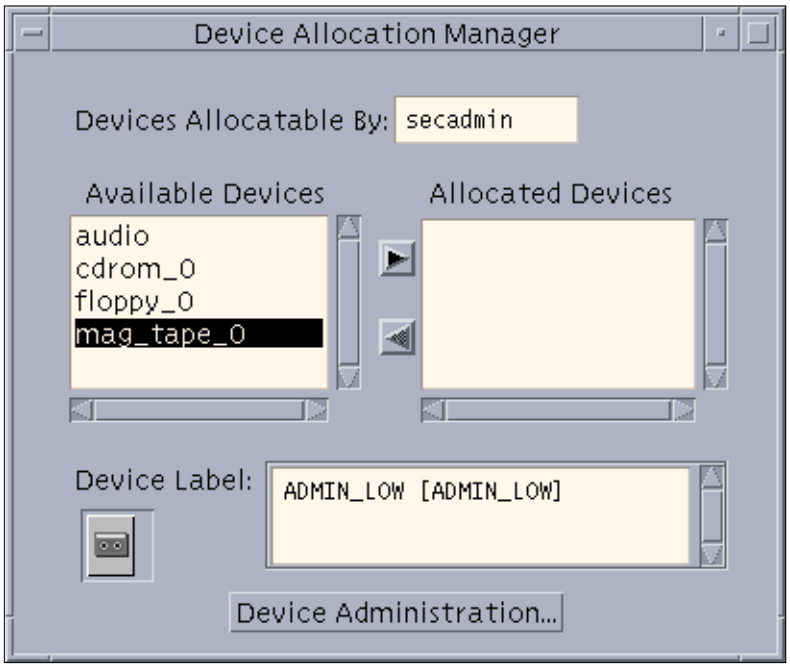



*Figure 15-10* Device Allocation Manager

3. **Click the Device Administration button to bring up the Device Allocation Maintenance dialog box shown in Figure 15-11 on page 422.**

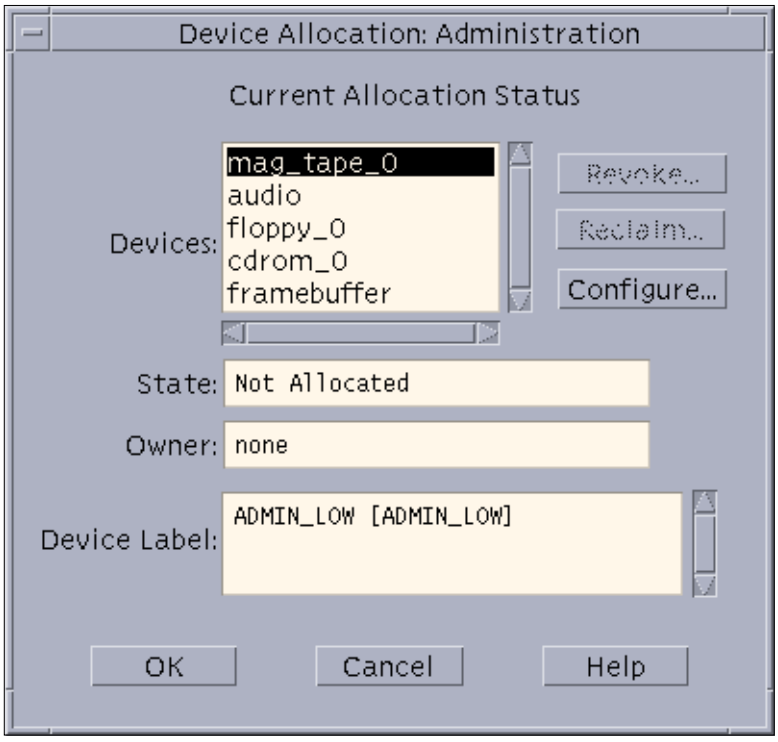



*Figure 15-11* Device Allocation: Administration Dialog Box

1. **Use the Administration dialog box to view the status of a device by highlighting the name of the device and checking the State: field.**
2. **If a device is in the Allocate Error State, click the Reclaim button to reclaim it.**
3. **If a device is allocated, either notify the owner to deallocate it or click the Revoke button to forcibly deallocate it.**
4. **Click OK to save the changes and close the dialog box.**

## ▼ To Add a New Allocatable Device or Make an Existing Device Allocatable

Follow the instructions in the *Installing Device Drivers* manual for Solaris, if needed, then do the following Trusted Solaris-specific steps.

1. **Assume the security administrator role and go to an ADMIN_LOW workspace.**
2. **Click the Application Manager icon in the Front Panel to open it.**
   See Figure 15-12.

*Figure 15-12* Application Manager Icon

3. **Double-click the System_Admin icon to access the administrative actions.**
4. **If no listing exists for the new device in** `/etc/security/device_maps`**, add the device's name along with a list of all the device special files associated with the device.**
   Use `dminfo(1MTSOL)` to check if the device is listed in the `device_maps(4TSOL)` file. See the `dminfo(1MTSOL)` man page and follow the format in the `device_maps(4TSOL)` man page.
   a. **Use the Admin Editor action to open the** `/etc/security/device_maps` **file for editing.**
   b. **Add a list of all the device special files associated with the device.**
      Shown is the entry for the mag_tape_0. Copy and modify it if you add another magnetic tape drive.

```
mag_tape_0:\
        st:\
        /dev/rst4 /dev/nrst4 /dev/rst12 /dev/nrst12 /dev/rst20
/dev/nrst20 /dev/rst28 /dev/nrst28 /dev/rmt/0 /dev/rmt/0n
/dev/rmt/0b /dev/rmt/0bn /dev/rmt/0l /dev/rmt/0m /dev/rmt/0h
/dev/rmt/0c /dev/rmt/0u /dev/rmt/0ln /dev/rmt/0mn /dev/rmt/0hn
/dev/rmt/0cn /dev/rmt/0un /dev/rmt/0ub /dev/rmt/0ubn:\
```

c. **Save and quit the file.**

5. **Use the Admin Editor action to create an empty lock file in** `/etc/security/dev`.

   a. **Open** `/etc/security/dev/`*devicename*, **with** *devicename* **the same as the name of the new device listed in the first field of the** `device_maps` **file.**

   b. **Without editing the file, save and quit it.**

   c. **Change the user and group of the file to bin, the mode to 000.**
   The lock file you create should have the same permissions, owner, group, and size shown in the long listing for the new empty lock file in the last line of the example.

```
$ chown bin device_name
$ chgrp bin device_name
$ chmod 000 device_name
$ ls -l | grep device_name
----------   1 bin      bin            0 Feb 24 11:58 device_name
```

6. **Create a device-clean script if needed for the new device.**
   If you add a Xylogics or an Archive tape drive, you can use or modify the `st_clean` script; otherwise, create your own. See “Device-Clean Scripts” on page 407.

7. **Create an entry for a new allocatable device in the** `/etc/security/device_allocate` **file.**

   a. **Use the Admin Editor action to open the** `/etc/security/device_allocate` **file for editing.**

   b. **Copy and modify some other entry for a similar device.**
   The example shows a copy of the entry for mag_tape_0.

```
new_device_name;st;0x00000000000000000000000000000000000000000000000000000000000000000;0x7fffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff;allocate device;/etc/security/lib/st_cle
an
```

   i. **Put the name of the device in the first field.**

   ii. **Put the device type in the second field.**

iii. **Leave the** `device_minimum` **and** `device_maximum` **label fields (fields 3 and 4) unmodified.**

iv. **Leave the name of the required authorization(s) in field unmodified.**

v. **Put the name of the** `device_clean`**(1MTSOL) script in field 6.**

c. **Save and quit the file.**

8. **To change the label range from the default range, to change the device allocation authorization, or to change whether the device is allocatable or whether allocation requires an authorization, use the Device Allocation Manager.**

   a. **Assume the security role in an ADMIN_LOW workspace, and 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 Trusted Desktop subpanel in the Front Panel. See Figure 15-10 on page 421.

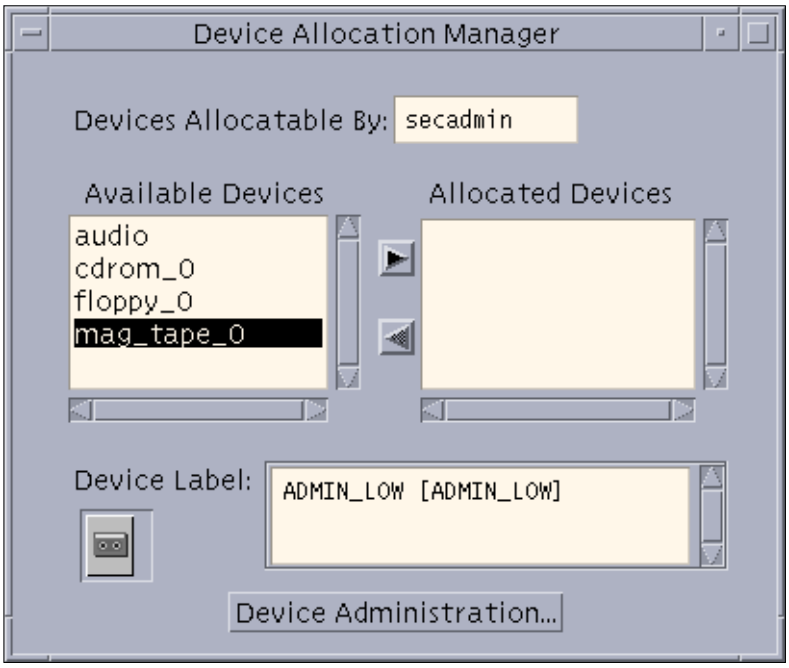



*Figure 15-13* Device Allocation Manager

**b. Click the Device Administration button to display the Device Allocation: Administration dialog box.**
See Figure 15-14 on page 427.

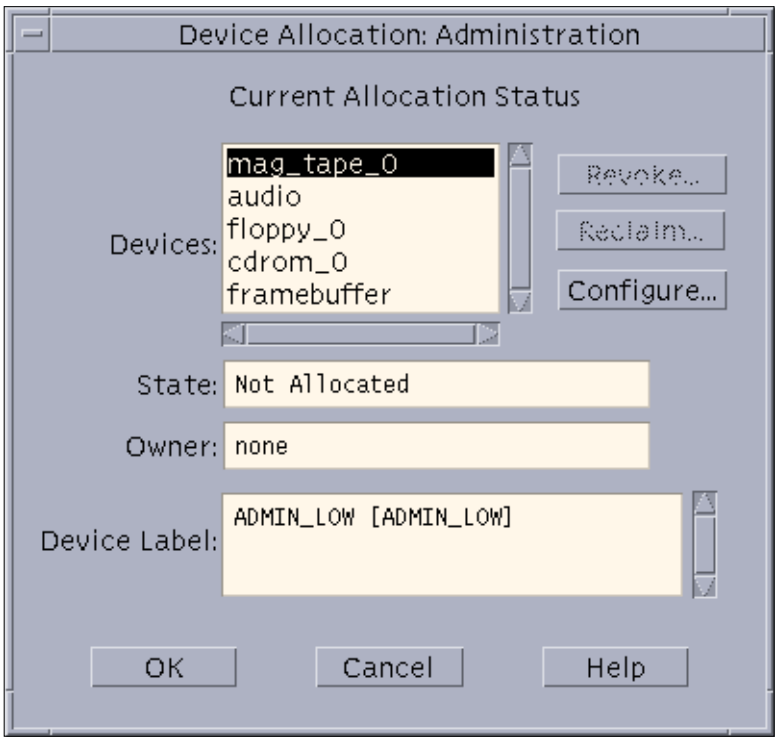



*Figure 15-14* Device Allocation: Administration Dialog Box

**c. Highlight the name of the device in the Devices list and click the Configure button to display the Device Allocation: Configuration dialog box.**
See Figure 15-15 on page 428.

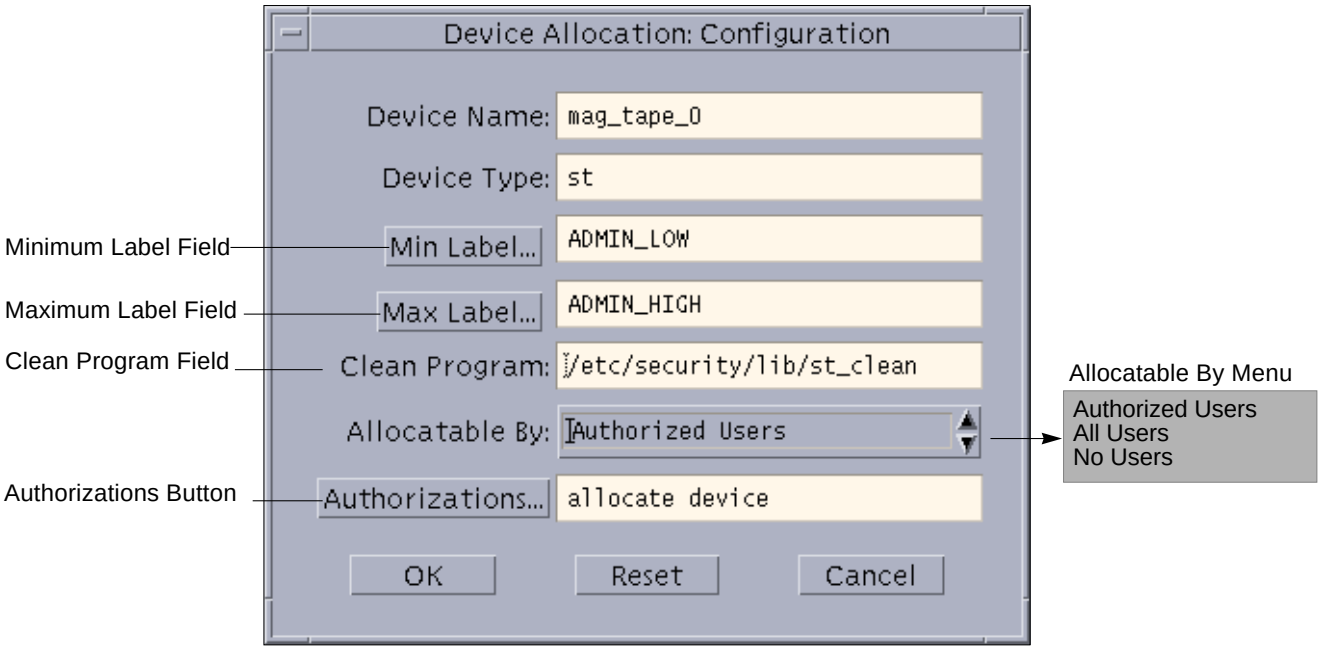



*Figure 15-15* Device Allocation: Configuration Dialog Box

**d. Change the label range if desired from the default range of ADMIN_LOW to a ADMIN_HIGH by clicking the Min Label and Max Label buttons and using the label builders that display to select the desired sensitivity label.**

**e. Change the name of the `device_clean(1MTSOL)` script as desired by editing the name in the text entry field to the right of Clean Program.**

**f. Change who can allocate the device from the default of Authorized Users by clicking the arrows to the right of the Allocatable By menu to change to All Users or No Users as desired.**

**g. Add a new device allocation authorization, if desired.**
To add a new authorization, see Chapter 2, “Miscellaneous Tasks and Procedures,” under “Extending Extendable Security Mechanisms” on page 31. After a new authorization is added it appears in the Not Required list of the Device Allocation: Authorizations dialog box.

**h. Change the authorizations as desired from the default of *allocate device* by clicking the Authorizations button to display the Device Allocation: Authorizations dialog box.**
See Figure 15-16.

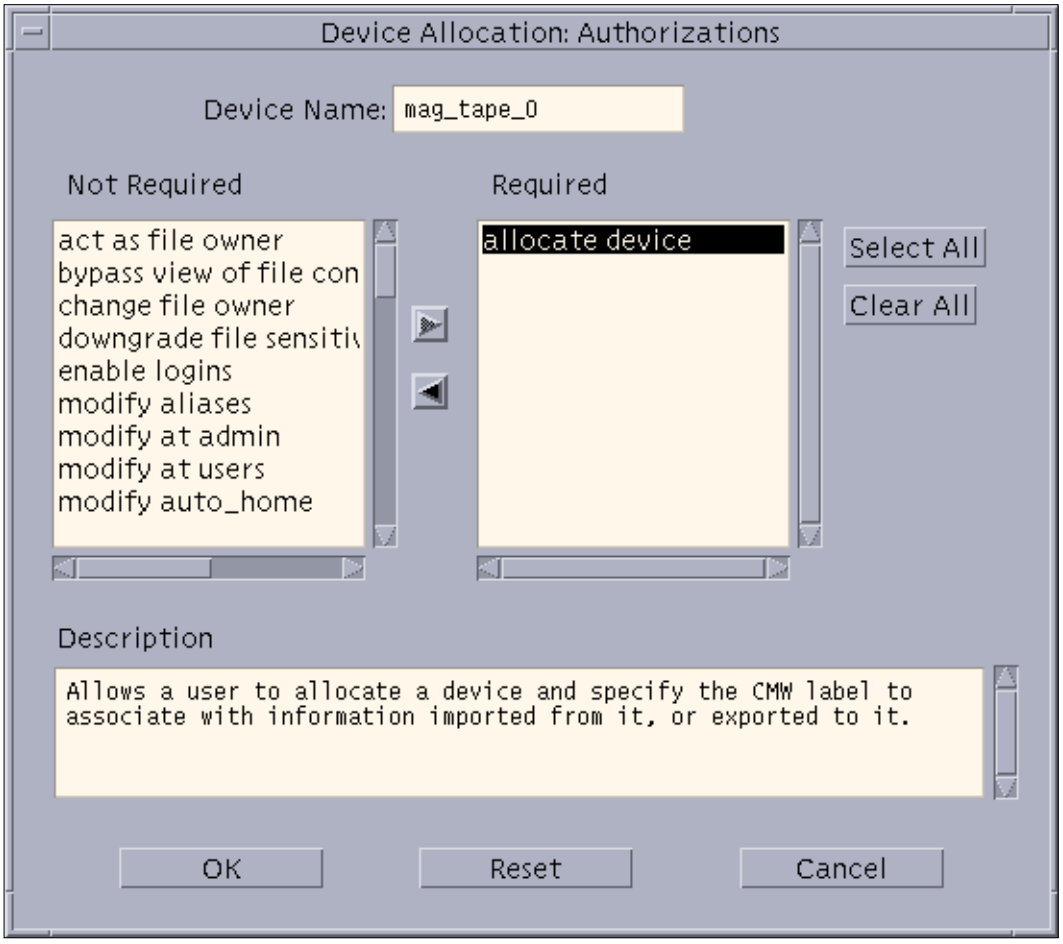



*Figure 15-16* Device Allocation: Configuration Dialog Box

**i. Change the default allocate device authorization, if desired, by selecting it in the Required list and using the arrows to move it to the Not Required List, and then selecting and moving another authorization from the Not Required list to Required.**

ii. **Add multiple authorizations to the Required list. if desired, by moving them from the Not Required to the Required list.**

i. **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.**

j. **If desired, configure a non-default policy for the device. See the `device_policy(4TSOL)` man page for the default policies and options.**

## ▼ To Add a New Non-Allocatable Device

Follow the instructions in the *Installing Device Drivers* manual for Solaris, if needed, then do the following Trusted Solaris-specific steps.

1. **Assume the security administrator role and go to an ADMIN_LOW workspace.**
2. **Click the Application Manager icon in the Front Panel to open it.** See Figure 15-12.

*Figure 15-17* Application Manager Icon

3. **Double-click the System_Admin icon to access the administrative actions.**

4. **For a device newly-added to the system, if no listing exists for it in** `/etc/security/device_maps`**, add a list of all the device special files associated with the device.**
   Use `dminfo(1MTSOL)` to check if the device is listed in the `device_maps(4TSOL)` file. See the `dminfo(1MTSOL)` man page and follow the format in the `device_maps(4TSOL)` man page.

   The framebuffer entry in `device_allocate(4TSOL)` can be most easily adapted, because the framebuffer is a non-allocatable device:.

```
framebuffer:fb:/dev/fb:
```

   The example shows an entry for a printer named tsolE connected on serial port `/dev/ttya`.

```
tsolE:lp:/dev/ttya:
```

5. **Save and quit the file.**

6. **Use the Admin Editor action to create an empty lock file in** `/etc/security/dev`.

   a. **Open** `/etc/security/dev/`*devicename*, **with** *devicename* **the same as the name of the new device listed in the first field of the device_maps file.**

   b. **Without editing the file, save and quit it.**

   c. **Change the user and group to bin, the mode to 000.**
   The example shows a lock file being modified for a printer named tsolE. The printer lock file you create should have the same permissions, owner, group, and size shown in the long listing for the new empty lock file in the last line of the example.

```
$ chown bin tsolE
$ chgrp bin tsolE
$ chmod 000 tsolE
$ ls -l | grep tsolE
----------   1 bin      bin            0 Feb 24 11:58 tsolE
```

7. **Create an entry for the new device in the** `/etc/security/device_allocate` **file.**

   a. **Use the Admin Editor action to open the** `/etc/security/device_allocate` **file for editing.**

   b. **Create an entry in** `device_allocate(4TSOL)` **by copying and modifying some other entry for a similar device.**

      The example shows the framebuffer entry cloned and renamed (the framebuffer is another non-allocatable device)

```
tsolE:lp:0x000...:0x7fff...;*;/bin/true
```

   c. **Put the name of the device in the first field.**

   d. **Put the device type in the second field.**

   e. **Leave the** `device_minimum` **and** `device_maximum` **label fields (fields 3 and 4) unmodified.**

   f. **Leave the asterisk (*), which means the device is not allocatable, in the required authorization(s) in field 5.**

   g. **Leave the name of the** `device_clean(1MTSOL)` **script in field 6 as** `/bin/true`**.**

   h. **Save and quit the file.**

8. **To change the label range from the default range of ADMIN_LOW to ADMIN_HIGH, use the Device Allocation Manager.**

   a. **As admin in an ADMIN_LOW workspace, 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 Trusted Desktop subpanel in the Front Panel.

   b. **Click the Device Administration button to display the Device Allocation: Administration dialog box.**

   c. **Highlight the name of the new device in the Devices list.**

   d. **Click the Configure button to display the Device Allocation: Configuration dialog box.**
   **See Figure 15-18 on page 433.**

e. **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 sensitivity label.**

f. **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.**

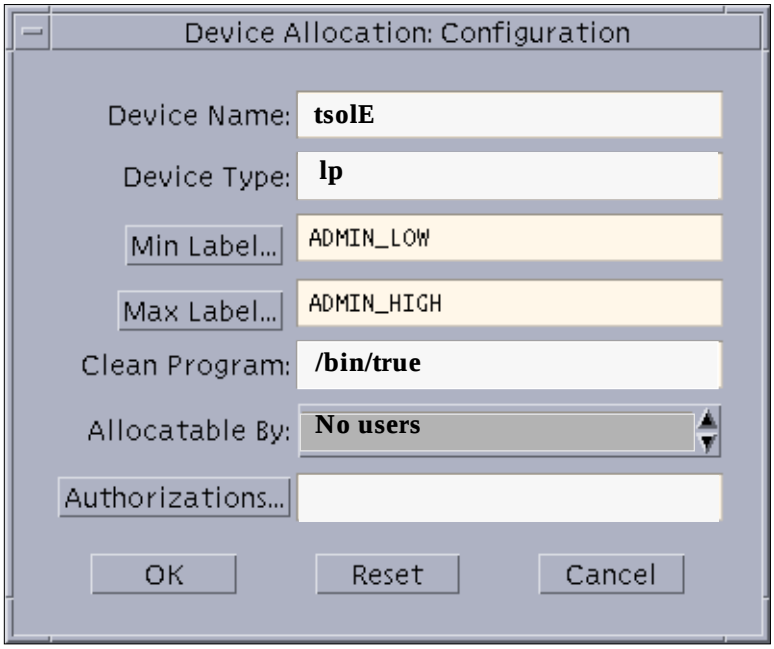



*Figure 15-18* Device Allocation: Configuration Dialog Box

## ▼ To Change or Add a Device Clean Script

1. **Write the script so that all usable data is purged from the physical device and that it returns 0 for success.**
2. **For devices with removable media, 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.**
3. **Put the script into** `/etc/security/lib`**.**

4. **Use the Device Allocation Manager to specify the new script for a device.**

   a. **As admin in an ADMIN_LOW workspace, 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 Trusted Desktop subpanel in the Front Panel.

   b. **Click the Device Administration button to display the Device Allocation: Administration dialog box.**

   c. **Highlight the name of the device in the Devices list to which you want to assign the new script.**

   d. **Click the Configure button to display the Device Allocation: Configuration dialog box.**
   **See Figure 15-18 on page 433.**

   e. **Change the name of the** `device_clean(1MTSOL)` **script as desired by editing the name in the text entry field to the right of Clean Program.**

   f. **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.**

# Adding Software 16

Security administrators can add the following types of software:

- Sun unbundled products and third-party applications that neither understand nor enforce Trusted Solaris security policy
- New programs, created by in-house developers 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)
- Shell scripts (created or approved by the security administrator)

Security administrators may also add to or modify the commands that run during start-up in run control scripts.

This chapter summarizes how the creation and use of commands, actions, and scripts is different in the Trusted Solaris system than it is in the base Solaris system. This chapter reviews how privileges are used by commands and actions and provides guidance on how the security administrator does the following:

- Brings in new software
- If the software needs privileges in order to run, finds out what privileges the software needs
- Decides whether giving any needed privileges to the new software is consistent with the site's security policy, and
- Makes privileges available to the software
- Makes privileges available to commands in run control scripts

# 16

See the *Trusted Solaris Developer's Guide* for more about how programs use privileges.

This chapter covers these main topics:


| | |
|---|---|
| *Controls for Software Creation and Use* | *page 438* |
| *Controls for Importing Software* | *page 438* |
| *Privileges* | *page 438* |
| *Alternatives to Assigning Privilege* | *page 440* |
| *Effects of the Execution Profiles on the Use of Commands and Actions* | *page 442* |
| *The Profile Shell, the System Shell, and Trusted Processes* | *page 443* |
| *Processes, Programs, and Their Privileges* | *page 445* |
| *Why Inheritable Privileges Are Important* | *page 450* |
| *How Privileges Are Assigned to Commands and Actions* | *page 453* |
| *Why Privileged Programs Need to Use Trusted Shared Libraries* | *page 454* |
| *Starting Commands With or Without Trusted Solaris Security Attributes During Boot* | *page 457* |
| *Security Administrator's Tasks in Adding Software* | *page 460* |
| *Issues Around the Adding of Privileges to Any Software* | *page 460* |
| *When Adding Existing Programs* | *page 461* |
| *When Adding a New Trusted Program* | *page 464* |
| *When Adding Actions* | *page 465* |
| *Modifying the Front Panel and Workspace Menu* | *page 467* |
| *Creating and Using Shell Scripts* | *page 469* |


This chapter provides these procedures:


| | |
|---|---|
| *To Add A Package from a CD-ROM* | *page 472* |
| *To Set Up an Application to Run WIth a Real UID of Root* | *page 473* |
| *To Set Up An Application to Run With An Effective UID of Root* | *page 473* |
| *To Find Out Which Privileges an Application Needs* | *page 473* |
| *To Give Forced Privileges to a Command* | *page 476* |
| *To Allow Trusted Programs to Link to Trusted Libraries* | *page 478* |
| *To Write a Profile Shell Script That Runs Privileged Commands* | *page 479* |

| | |
|---|---|
| *To Write a Standard Shell Script That Runs Privileged Commands When Executed in a Profile Shell* | *page 480* |
| *To Specify Commands to Run With or Without Special Attributes During Boot* | *page 481* |
| *To Restore Privileges Lost When a File is Edited* | *page 483* |


## *Review of Terms and Concepts*

As configured in the default system, the security administrator role is the only account that is able to do the following:

- Import and export software at multiple sensitivity labels
- Install software at ADMIN_LOW in the public directories (such as `/etc` and `/usr/bin`) that allow use of the software by multiple users at all sensitivity labels.
- Assign privileges to program files, although this ability may be given to other accounts by assigning the needed authorization.
- Use the Profile Manager to assign privileges that are in effect when a command or action is executed in a trusted process from which it can inherit privileges.

---

**Note –** Because applications, whether they are externally or internally obtained, and shell scripts are added to a site's execution profiles as commands, the term *command* is used frequently in this chapter when referring to applications, site-developed executable programs, and shell scripts.

---

See "Privileges," and the following sections, which define what it means for a command to have privileges and for a command or action to inherit privileges.

Assuming that a site has procedures to screen imported software for viruses and worms, the security administrator, in most cases, does not need to be consulted before programs or scripts are imported, created or used if the software meets the following criteria:

- It does not need to run with privilege
- It does not need to run with an effective UID or GID that differs from the real UID or GID of the invoking user
- It does not need to run at multiple labels

## Controls for Software Creation and Use

The security administrator controls beforehand *who can create* programs, actions, and shell scripts by controlling what commands and actions are made available to individual accounts through the profile mechanism. For example, if a user or role account is not allowed to use the Create Action action or some other tool that would allow the editing of files that create an action, that account simply cannot create an action. And, even if an account can create an action, the account would not be able to use any new action that he or she might create, without the security administrator's cooperation, unless the account already has the All profile, which turns off all checks for the use of commands or actions. To allow anyone without the All profile to use a new action, the security administrator would have to add the action to site's profiles. The All profile also turns off all checks for commands, and would enable an account to run any standard shell and enter any existing or new command or script.

Programs and shell scripts are controlled in a slightly different manner than actions. If an account is allowed to use a terminal or an application that allows him or her to create a new program or shell script, the resulting new program or shell script can be executed by the creator or by anyone who has access to the software and who has a standard UNIX shell. For accounts (such as secadmin and admin accounts) that have the profile shell and no other shell assigned, the security administrator would need to add the name of the new command to one of the account's profiles in order for the account to use it, unless the account also has the All profile.

## Controls for Importing Software

Similar mechanisms are in place for controlling the importing of programs, actions and scripts as there are for creating them. The security administrator controls *who can bring in software* by granting or denying the device allocation authorization to an individual. If an account has the device allocation authorization, the account is then restricted to importing or exporting data at a single sensitivity label within that user's clearance.

## Privileges

Having privilege means having the capability to override some aspect of security policy.

In standard UNIX systems, only commands running with the user ID of 0 can run with all the powers of the superuser, while commands running with any other user ID have none. In the Trusted Solaris system, however, a command with any user ID may be configured to use some but not all of powers that are reserved for the superuser in standard UNIX system. The ability of the UNIX super-user to bypass access restrictions, to execute restricted commands, and to use some command options not available to other users has been replaced with privileges. The superuser in standard Solaris and other UNIX operating systems has all privileges in the system, including the ability to bypass all DAC restrictions. The privileges mechanism in Trusted Solaris breaks the power of the superuser into many small pieces and provides additional power that may be needed to bypass MAC restrictions. Providing discrete privileges makes it possible to assign to programs only those privileges they need to do their work and no more.

## *Required Privileges*

When a command or one of its options or an action needs a privilege in order to succeed, that privilege is a *required* privilege; if a required privilege is not available to a command, option or action, it will not work at all. Required privileges are indicated on the man page with the words “must have” as shown in this sentence: "The `ifconfig(1MTSOL)` command must have the sys_net_config privilege to modify network interfaces."

## *Override Privileges*

When a command or action is designed to work within security policy, and then it fails when certain DAC or MAC checks are not passed, an *override* privilege may be assigned at the security administrator's discretion. On man pages, the names of privileges that may be used to override access restrictions are given in the ERRORS section.

The DAC override privileges are file_dac_read and file_dac_write. The MAC override privileges are file_mac_read and file_mac_write. If a user does not have DAC or MAC access to a file, the security administrator may assign one or both of these privileges to a command or action, depending on whether MAC or DAC applies and on whether read or write access or both are desired.

## Alternatives to Assigning Privilege

Besides being able to assign an override privilege to a command or action to make it work, the security administrator has other options:

- Assign an effective UID
- Assign an effective GID
- Assign a sensitivity label at which the command executes

For example, to avoid the use of privilege the security administrator may configure a command or action to execute with another user's ID or an alternate group ID that allows access to the file or directory based on its permissions or its ACL. When software executes with another UID or GID than the *real UID or GID* of the person who invoked it, it is said to be running with an *effective UID or GID*. For another example, the command or action may be specified to run at a sensitivity label that removes the need to give it a MAC override privilege.

## Principle of Least Privilege

The *principle of least privilege* requires that each command or action in the system be granted the most restricted set of privileges that it needs to do the required task for the person who is using it. The principle of least privilege is supported by making privileges available to programs on an as-needed basis, and by using privilege bracketing, which is used within programs to turn the use of the program's privileges on and off so that privileges are only in effect while they are being used for a specific purpose during the run time of the program. See the *Trusted Solaris Developer's Guide* for more about privilege bracketing.

## File Privilege Sets

Executable program files can have two *file privilege sets* associated with them: *forced* and *allowed*. The *allowed* privilege set determines which privileges *may* be made available to a program, while any privileges in a program file's *forced* privilege set are always made available to the program, as long as the privileges are also in the allowed set, no matter what shell the program is executed in or who invokes it.

## How Two Standard Programs Use Privilege in Trusted Solaris

The ways that `ls(1)` and `mount(1MTSOL)` use privileges illustrates some of the principles talked about so far in this chapter.

`ls(1B)` is an example of a well-known Trusted Solaris program that does not need privileges in order to work, as long as it is used to list files that DAC and MAC restrictions allow it to see. Therefore, `ls` has no *required* privileges. However, the security administrator could assign *override* privileges if he or she wants to allow `ls` to bypass DAC or MAC security policy for some reason.

When executed by a normal user for the purpose of seeing which file systems are mounted on that user's machine, `mount(1MTSOL)` needs no privileges. When used to mount file systems, `mount` always needs the privilege sys_mount (so sys_mount is a *required* privilege for `mount`). Depending on whether it is being used to mount file systems remotely or to specify mount-time security attributes, `mount` also may need several other *override* privileges. In the default configuration, the executable program file for `mount` is specified with all allowed privileges and no forced privileges. The System Management profile (which is assigned to the system administrator role) has mount defined with all the privileges `mount` needs to inherit when mounting file systems. See the `mount(1MTSOL)` man page for more details of mount's privilege requirements.

## Actions

Actions are a feature from the CDE window system that are also used extensively in the Trusted Solaris window system, with a number of restrictions on their use and creation that are necessary to make them compatible with the Trusted Solaris security policy. The restrictions on their use and creation are described in "When Adding Actions" on page 465.

Actions are instructions that are written to automate tasks such as opening files for editing or running applications. Actions are defined much like application macros or programming functions. Each action is defined with a name that is used to run the action. Actions can be attached to icons, Front Panel controls, and menu items.

# 16

## *Effects of the Execution Profiles on the Use of Commands and Actions*

The security administrator may *enable* an account to bypass system security policy by specifying one or more of the following in the account's execution profiles:

- Authorizations
- Privileges assigned to specific commands or actions
- Effective UIDs or GIDs assigned to specific commands or actions

The privileges specified in execution profiles are made available to commands and actions by inheritance. Commands can inherit privileges and can run with an effective UID or GID or a sensitivity label specified in one of the invoking account's profiles only when they are invoked in a *profile shell* or in a *system shell*. Actions can inherit privileges and can run with an effective UID or GID or a specific sensitivity label only when they are launched from one of the trusted processes in the window system (which are described in "The Profile Shell, the System Shell, and Trusted Processes" on page 443).

A user account may be given the All profile that bypasses all checks on the use of actions or commands, and then the account can use *all actions or commands without inheritable privileges*—giving that account the freedom of access available to a standard UNIX user.

If an account has a standard UNIX shell, either as the default shell or listed in one of its profiles, the user can execute *all commands* in that shell (without inheritable privileges) but may not be able to launch all actions, depending on how actions are configured in the account's profiles. A user or role account is *restricted* to use only the set of actions that are explicitly specified in that account's execution profiles (unless one of the profiles is the All profile). Commands are handled differently; only if an account is given a profile shell as the default shell is the account is restricted to use only those commands that are listed in the account's profiles

Each account may have multiple profiles. The order in which profiles are assigned is important. The profile shell and trusted processes in the window system search the profiles in the order specified for the account in the User Manager. If more than one profile is assigned to an account, the order is significant because the first time the profile shell or trusted process finds the command or action in any profile, the command or action is given whatever security attributes have been assigned to it in the first profile where it appears.

For example, Table 16-1 shows user account roseanne has All, ProfileA, ProfileB, and ProfileC, in that order, in its set of profiles.

*Table 16-1* Example of an Incorrect Ordering of Profiles in an Account's List

| Account name | Default Shell | Profiles | Command |
|---|---|---|---|
| roseanne | pfsh | All | all commands with inheritable=none |
| | | ProfileA | `command1` with inheritable=1,2,3 |
| | | ProfileB | `command1` with inheritable=2,3,5 |
| | | ProfileC | `command1` with inheritable=7,9, 12 |

If roseanne executes `command1`, the profile mechanism finds the All profile first, which allows the use of any command without special attributes, and stops there, allowing roseanne to execute the command without giving it privileges or other special attributes. If the task that roseanne needs to do requires that `command1` has privileges 7, 9, and 12 in order to succeed, the security administrator should move ProfileC to the top of roseanne's list of profiles.

If the All profile is assigned to an account, it should be at the bottom of an account's list of profiles, where it can act as a type of fallback mechanism for commands, allowing the account to use any command without privileges, effective UID or GID or specified sensitivity label, if the command is not otherwise explicitly specified in one of the account's other profiles.

## *The Profile Shell, the System Shell, and Trusted Processes*

The profile shell, `pfsh(1MTSOL)`, is the default shell for the security administrator and the system administrator roles, and the `pfsh` may be assigned, at the discretion of the security administrator, as the default shell for any other user or role account. Any account with the profile shell as its default shell is restricted to use only *commands* that are *specified in the account's execution profiles*.

The system shell, `sysh(1MTSOL)`, is a shell that is used to control the use of privileges by commands run from *run control* (`rc`) scripts. `sysh` allows any command to execute but consults profiles for any privileges, effective user ID, effective group ID, and sensitivity label with which the command is to be run. See "Starting Commands With or Without Trusted Solaris Security Attributes During Boot" on page 457 for more about `sysh`.

The window system's trusted processes are:

- The Front Panel
- Subpanels of the Front Panel
- The Workspace Menu
- The File Manager and
- The Application Manager

Trusted processes in the window system are available to everyone, but accounts are restricted to be able to access from the window system only the *actions* that are *specified in the account's execution profiles*. For example, the administrative actions that are used for editing configuration files, which are in the System_Admin folder in the Application Manager, can only be used if they are in one of the account's profiles. Therefore, because Edit Encodings is in the Object Label profile assigned to the security administrator role, he or she can use the Edit Encodings action, but he or she cannot use the Set Mount Points action because it is not in any of the role's profiles.

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 comes up 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 profiles when the account attempts to invoke a command, `Xtsolusersession` also consults the account's profiles when the account attempts to launch an action. In either cases, if the action is in the user's profiles, it is run with any specified privileges or an effective UID or GID if those are specified.

The profile shell and the window manager run with all privileges, because they need to pass any inheritable privileges specified for a command to the command when it is executing.

# Processes, Programs, and Their Privileges

A process is derived from a file containing an executable program. At any instant, a process is always executing a program. A process is created using the `fork(2)` system call, which creates a duplicate of the calling process. The new process inherits all the parent's attributes, including the currently executing program. A process can change the program it is executing with the `exec(2)` system call, which replaces the entire process address space with new versions derived from the program file named in the `execve` call.

For example, the process running the profile shell, `pfsh`, uses `execve` to execute the `mount` command.

## Process Privilege Sets

Four privilege sets are associated with each process:

- Permitted Privileges

  The privileges in a process's *permitted privilege set* are those the process is *allowed* to use.

- Effective Privileges

  The privileges in a process's *effective privilege set* are those a process is *currently using*. System calls that check for privilege check a process's set of effective privileges.

- Inheritable Privileges

  The privileges in a process's *inheritable privilege set* are those a process *may pass to another process* when another program is invoked with an `exec(2)`.

- Saved Privileges

  The privileges in a process's *saved privilege set* contains the set of privileges *actually inherited* across an `execve`. The contents of the saved privilege set become invalid if a process changes its effective user ID.

When a process executes a program through the `execve(2TSOL)` system call, the permitted (P) and effective (E) privilege sets are reset equal to the same value, which is the intersection of the process' previously existing inheritable (I) privileges and the program file's allowed (A) privileges intersected with the program file's forced (F) privileges:

```
P=E=(I[process] union F[program] restricted by A[program])
```

The saved privilege set is set initially to the intersection of the existing inheritable privilege set and the file's allowed privileges:

```
S=(I[process] intersected by A
```

*Keeping the saved privilege allows the process to determine which privileges it had when the currently executing program was invoked.*

When a new program is invoked, the inheritable privilege set is initially set to be the same as the inheritable privileges of the process that invoked the current program:

```
I[new]=I[old]
```

See “Why Inheritable Privileges Are Important” on page 450 for the benefits of setting the inheritable privileges without reference to the forced or allowed privileges on an executing program.

If the effective UID is set by `setuid(2TSOL)`, to be different from the original, the effective set is copied to the saved set and the effective set is cleared:

```
S=E; E=0
```

If the process changes its effective user ID back to the original, the saved privilege set is copied to the effective set, thus restoring its privileged state:

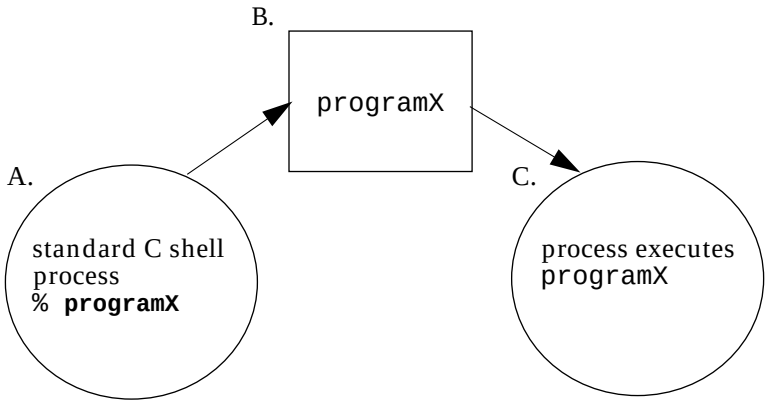
```
E=S
```

## Examples of How Processes Acquire Privileges

The following examples show how a process acquires privileges in a standard UNIX shell compared to how a process acquires privileges in a profile shell when a program is executed. The first example shows that when a program executes in a standard UNIX shell, the program's process can use only privileges that are in the program's forced and allowed privilege sets. The second example shows that when a program executes in a profile shell, the program's process can acquire privileges from the shell's inheritable set, which it can use if the privileges are also in the allowed privilege set of the program itself. In the examples, for brevity, the privileges are assigned numbers instead of names.

### In a Standard Shell

The circle on the left side of Figure 16-1 stands for a standard UNIX shell's process in which the user enters the command `programX`. The inheritable privilege set shown for the shell's process is null because standard shells' processes have no privileges.



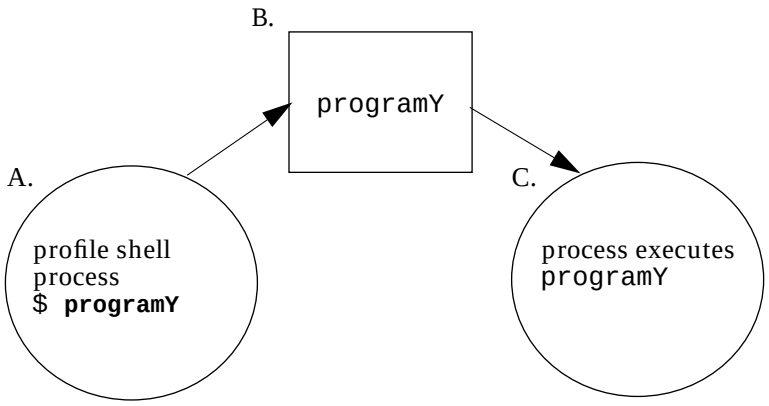

A.
standard UNIX shell process with inheritable=none,
B.
`programX` file's privilege sets: forced=1,3,5; allowed=1,3,5,11,12,19
C.
process running `programX` now has permitted=1,3,5; effective=1,3,5; inheritable=none

*Figure 16-1* Process Acquiring Forced Privileges When Run in a Normal User's Shell

In Figure 16-1 on page 447, the square in the middle stands for the `programX`, which has forced and allowed privileges as shown. The forced privileges are 1, 3, and 5. The allowed privileges are 1, 3, 5, 11, 12, and 19. The circle on the right stands for the process executing `programX` in which the permitted and effective privileges sets are given the forced privileges associated with the program file. Figure 16-1 illustrates that when the parent process is a standard shell, and therefore it has no inheritable privileges, the privilege sets associated with a new process contain only the forced privileges acquired from the program file, and the inheritable set of the new process remains empty.

## *In a Profile Shell*

In Figure 16-2, a user or a role account executes `programY` in a profile shell.




A.
In one of the invoking account's profiles, `programY` has inheritable=10,12,19, so `pfsh` sets its own inheritable=10,12,19 and executes `programY`.
B.
`programY` file's privilege sets: forced=none, allowed=10,12,19
C.
process running `programY` now has inheritable=10,12,19; effective=10,12,19; permitted=10,12,19

*Figure 16-2* Process Inheriting Privileges From the Profile Shell

In Figure 16-2 on page 448, the profile shell checks which user or role is executing the command, checks to make sure the command is in one of the account's profiles, checks to see if any privileges have been specified for the

command in any of the account's profiles, and puts only the specified privileges into its own inheritable set. As shown in the illustration, the executable `programY` itself has no forced privileges, and its allowed set includes 10, 12, and 19. Because `programY` has the three privileges 10, 12 and 19 in its allowed set, the process running `programY` can inherit privileges 10, 12 and 19 from the parent process, and privileges 10, 12, and 19 are put into the process's effective and permitted sets.

If there are no privileges in a program file's allowed set, then the permitted and effective sets for the process running the program always will be empty.

## *How a Process Executing the* `mount` *Command Acquires Privileges*

This section shows how the `mount` command used in the example in "How Two Standard Programs Use Privilege in Trusted Solaris" on page 441 is executed in a profile shell process and acquires privileges. When `mount` is executed in the profile shell, the process running `pfsh` does the following:

- It checks that the invoking account has the mount command in one of its profiles
- It puts the sys_mount privilege and the other privileges that are specified for mount in the invoking account's profiles into its own (`pfsh`'s own) inheritable set
- It runs `execve`, which creates a process running the `mount` command

In `mount`'s process, the *inheritable* privileges are set to be the same as the inheritable privileges of the profile shell's process that invoked `mount`. Then the forced and allowed privileges on `mount`'s program file are used in conjunction with the inheritable set in determining the process's permitted, effective, and saved sets.

To determine the *permitted* and *effective* set, a process's inheritable privileges are first combined with any additional privileges from the forced set on the program file—which does not apply here since `mount` has no forced privileges. The combination of both the inheritable and forced privileges is limited to the allowed privileges specified for the program file, and because the `mount` program file has all allowed privileges, `mount` can use all of its inheritable privileges, and, as a result, the permitted and effective sets of the process running mount are set equal to the inheritable privilege set.

`mount`'s *saved* privilege set is set to be the inheritable set and the forced and allowed privileges on mount's program file.

## Why Inheritable Privileges Are Important

Inheritable privileges are discussed in detail under "Process Privilege Sets" on page 445. Inheritable privileges are important for security administrators because privilege inheritance is used by:

- The profile mechanism to pass privileges to commands invoked in the profile shell
- The system shell to pass privileges to commands invoked in the system shell]
- Trusted processes in the window system to pass privileges to actions

As described in "Process Privilege Sets" on page 445, 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. Maintaining the inheritable set without reference to the program file's forced or allowed set has the following two effects:

- 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.

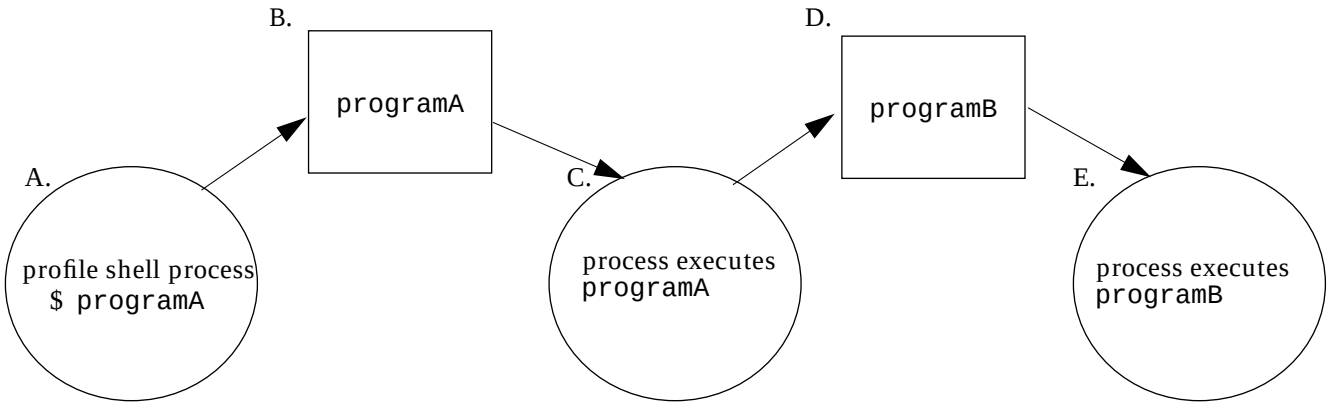
  See "When a Program File Has No Allowed Privileges" for details.

- The benefit of setting I[new]=I[old] without reference to forced privileges is that forced privileges cannot be used by shell scripts.

  See "When a Program File Has No Forced Privileges" on page 451 for details.

## When a Program File Has No Allowed Privileges

When a program file has does not have allowed privileges, the inheritable set of the process executing the program is not reduced to match the allowed privileges on the 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. See Figure 16-3, "How a Program That Cannot Use Privileges Can Pass Them to A Program That Can."

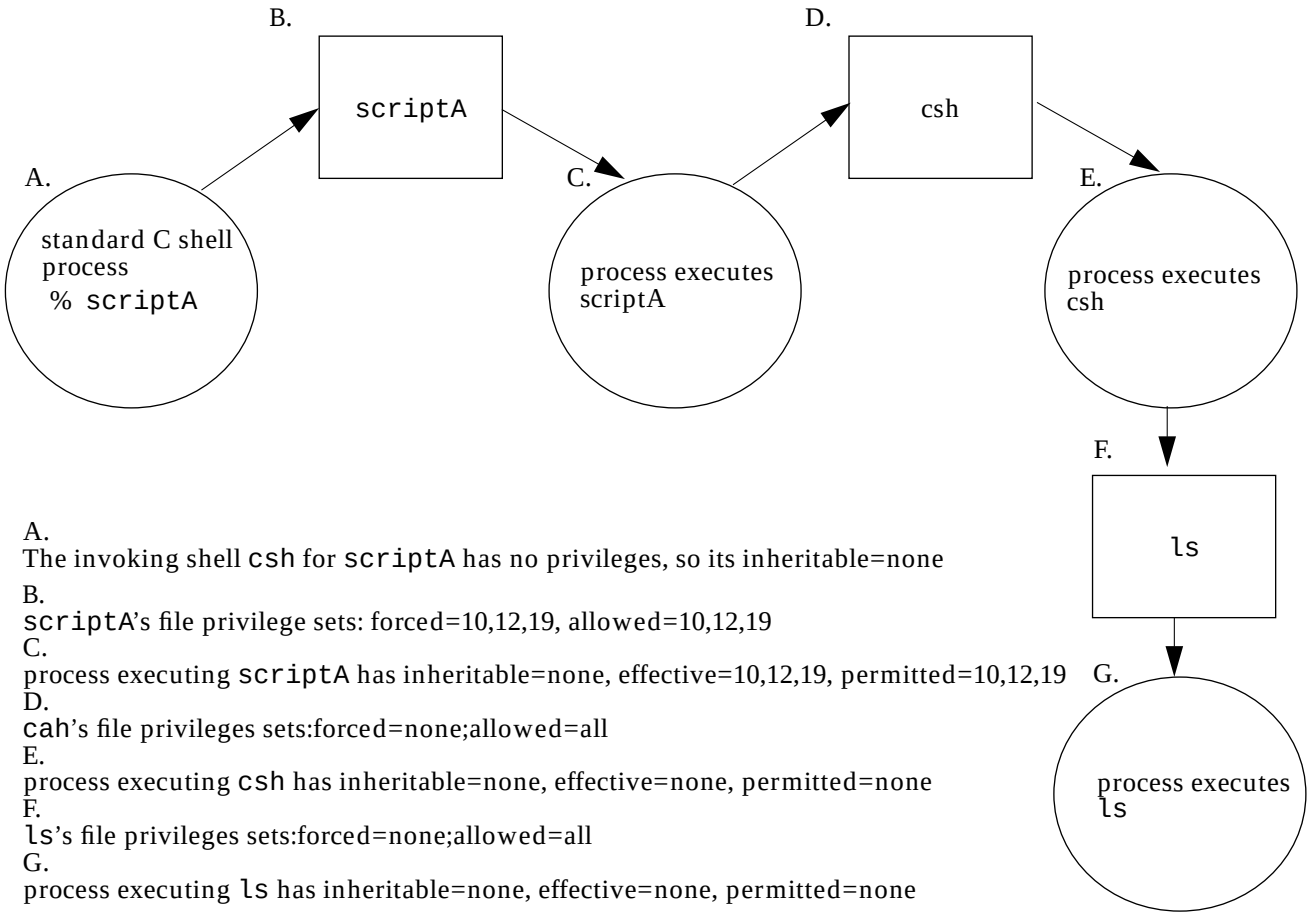



A.
In one of the invoking account's profiles, `programA` has inheritable=10,12,19, so `pfsh` sets its own inheritable=10,12,19 and executes `programA`.
B.
`programA` file's privilege sets: forced=none, allowed=none
C.
process executing `programA` has inheritable=10,12,19; effective=none
D.
`programB` file's privileges sets: forced=none;allowed=10,12,19
E.
process executing `programB` has inheritable=10,12,19;effective=10,12,19

*Figure 16-3* How a Program That Cannot Use Privileges Can Pass Them to A Program That Can

## *When a Program File Has No Forced Privileges*

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 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 Figure 16-4 on page 452, "How Forced Privilege Shell Scripts Are Prevented From Passing Forced Privileges to Their Commands."




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.
`cah`'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 16-4* How Forced Privilege Shell Scripts Are Prevented From Passing Forced Privileges to Their Commands

## How Privileges Are Assigned to Commands and Actions

Trusted Solaris commands and actions have already been assessed as described in this chapter and have been assigned privileges if any privileges are required for them to do their work. After a site is configured, a privilege should be granted by a site's security administrator only if the security administrator is convinced that the command or action itself or the person invoking the command or action will use the privilege in a trustworthy manner.

The security administrator makes privileges available by:

- Assigning forced privileges to the executable file itself (for commands only) or
- Making them inheritable by a command when it is invoked in the profile shell or when it is launched from a trusted process in the window system or by an action when it is launched from a trusted process in the window system.

### Giving Forced Privileges to a Command

The security administrator can assign forced privileges to an executable file for a command by using the File Manager—Privileges dialog box or by entering the `setfpriv(1TSOL)` command in a profile shell, as described under “To Give Forced Privileges to a Command” on page 476.

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 anyone from executing such a command with privilege would be by controlling access to the command itself—by giving an account the profile shell as its default shell and by not assigning the command or any terminal emulator in any of that account’s profiles.

To change the privileges on an executable file, the process’s sensitivity label must allow MAC write access to the file; DAC write permission is not required. The forced and allowed privilege sets of a file can only be changed either by the owner at the same sensitivity label (write-equal) or by a security administrator (as configured in the default system) in an ADMIN_LOW workspace (write-up).

To give more detail, the forced and allowed privilege sets of a file can only 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

See also the `setfpriv(1TSOL)` man page.

---

**Note –** If you assign forced privileges using the File Manager—Privileges dialog box, it automatically assigns the same set of allowed privileges. However, the `setfpriv` command does not allow you to set any forced privileges unless they are in the file's set of allowed privileges or unless you are setting the allowed and forced set appropriately in the same command line.

---

### *Giving Inheritable Privileges to a Command or Action*

The security administrator can specify inheritable privileges for a command or an action in an execution profile (using the Profile Manager) and assign the profile to a user or role account using the User Manager. See Chapter 5, "Using the User Manager to Configure User and Role Accounts," and Chapter 8, "Managing Execution Profiles for Users and Roles" for how to use the User Manager and Profile Manager.

---

**Note –** For privileges to be made available by inheritance, the privileges must be available in the command's allowed privilege set.

---

## *Why Privileged Programs Need to Use Trusted Shared Libraries*

Most applications use shared library routines. In Trusted Solaris, the security administrator needs to make sure that shared libraries used by any application that requires privilege(s) are in a trusted directory. The dynamic linking of privileged applications to shared libraries is restricted—to ensure that privileged applications can never use untrusted libraries. A privileged application that tries to link to an untrusted library fails with an error like: error: cannot load **** libraries.

## Default Trusted Shared Library Directories

The default trusted shared libraries are stored in the directories listed in Table 16-2..

*Table 16-2* Default Directories for Trusted Shared Libraries

| | |
|---|---|
| **Trusted C function libraries** | `/usr/lib` and `/etc/lib` |
| **Trusted extensions to X Server** | `/usr/openwin/lib` and `/usr/dt/lib` |

The directories listed in Table 16-2 are protected by mandatory and discretionary access controls to keep anyone but administrators from writing into them or modifying existing library files.

## Shared Libraries Used by Third Party or Site-Created Applications

If at all possible, when a third party or site-created application is given privilege(s) by a site's security administrator, any shared libraries on which the trusted application relies should be moved into one of the default trusted shared library directories shown in Table 16-2. Another alternative is to define a shared library's directory as trusted, but this alternative should not be used unless the libraries cannot be moved into one of the existing trusted directories. To identify a library directory as trusted, the security administrator lists the directory pathname for a privileged application's library in a `rtld` file in `/etc/security/tsol` .

---

**Note –** Any application that is given privilege becomes trusted. The security administrator must make sure that a program that needs privileges is actually worthy of trust. Because any application's libraries listed in `rtld` become trusted, they require the same level of protection as the default libraries. The security administrator should ensure that the MAC and DAC permissions on any directory listed in `rtld`are the same as they are on the default trusted libraries.

---

See the `ld(1TSOL)` man page for information on the link editor for object files and on the `rtld` file.

# 16

## *Example of Privileged Java Applications Using Java Libraries*

Like any other privileged applications, Java applications that are given privileges in order to work also need to have their shared libraries identified as trusted, as described in the previous two sections. Here is a request for help from a service representative:

```
I have a customer that is having problems running Java
applications with privileges. He gets errors:error can't load
java libraries
```

Because the site's security administrator cannot change the location of the library to which the application is dynamically linking, the security administrato needs to update `/etc/security/tsol/rtld` with the path of the Java libaries (`/usr/java/lib/` for example). See "To Allow Trusted Programs to Link to Trusted Libraries" on page 478.

# *Starting Commands With or Without Trusted Solaris Security Attributes During Boot*

As is the case in the base Solaris system, the commands that run during boot of the Trusted Solaris system may be added to or otherwise modified by administrative action.The base behavior is described in “Run Control Scripts” in the Solaris *System Administration Guide, Volume 1* and on the `init.d`(4) man page. See the README in each `/etc/rc`*n*`.d` directory for some guidelines about the numbering of the scripts that start system services.

During boot, the `/sbin/rc`*n* scripts execute the scripts in the corresponding `/etc/rc`*n*`.d` directories. The `number` *n* in the run control scripts’ names and run control directories’ names stands for the run level.

The scripts in the `/etc/rcn.d` directories are hard links to scripts actually located in `/etc/init.d`. For example, `sendmail` scripts with different names in the `/etc/rc0.d`, `rc1.d`, and `rc2.d` directories are actually hard links to the `/etc/init.d/sendmail` script. The three links to `/etc/init.d/sendmail` in the `/etc/rc`*n*`.d` directories are shown in Figure 16-5.

```
/etc/rc0.d/

    K57sendmail

/etc/rc1.d/

    K57sendmail

/etc/rc2.d/

    S88sendmail
```

*Figure 16-5* `/etc/initd.d/sendmail` Linked to `/etc/rc`*n*`.d` Directories

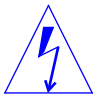
A script in the `/etc/rcn.d` directories starts with a S prefix when the script needs to be run with the `start` option, and it starts with a K prefix when it needs to be run with the `stop` option. In run levels 0 and 1, `sendmail` is stopped (as indicated by the K prefix on the filename), and in run level 2, sendmail is started (as indicated with the S prefix on the filename).

After installing a new script into `/etc/init.d`, the security administrator does the following:

- Identifies where in the boot sequence the script needs to be started and/or stopped
- In one or more `/etc/rc`*n*`.d` directory, makes a hard link to the script in `/etc/init.d`
  - Uses the proper prefix in the target file's name for either starting or stopping
  - Uses the proper numbers in the target file's name to help determine the order in which the script is executed during the run level

The point in the boot sequence at which a new script should be run also determines whether the security administrator puts the command run by the script into an execution profile stored in a local `tsolprof` file or into an execution profile stored in the `tsolprof` NIS+ map, as discussed in more detail in "To Specify Commands to Run With or Without Special Attributes During Boot" on page 481.

In the Trusted Solaris system, the `/sbin/rc`*n* scripts have been modified to use the system shell, `sysh(1MTSOL)`, instead of the Bourne shell, `sh(1)`. In the default Trusted Solaris system, a *boot* execution profile has been set up to specify security attributes for commands started during boot. In the `/sbin/rc`*n* scripts, `/bin/sysh` is used without a profile name argument because it looks at the boot profile by default.



**Warning –** Do not modify the boot profile or the `/sbin/rc`*n* scripts.

The security administrator should follow the steps in "To Specify Commands to Run With or Without Special Attributes During Boot" on page 481 to create new profile that has the name of the command run by the script and should use the `sysh` shell in the script, telling the `sysh` to consult the new profile by using `sysh`'s `setprof` option.

## *Using Scripts in the* `/etc/init.d` *Directory to Start and Stop Services*

In Solaris, the superuser can start and stop any command (also referred to a service) in the `/etc/init.d` directory by entering the name of the command followed by either the `start` or `stop` options. For example, the superuser can enter:

```
# /etc/init.d/sendmail start
and
# /etc/init.d/sendmail stop
```

*Figure 16-6* Starting and Stopping sendmail Using the `start` and `stop` Options with the `/etc/init.d/sendmail` Script

Stopping `sendmail` and other commands in a Trusted Solaris system as shown in Figure 16-6 does not require privileges, but the command must be run by an administrative role in an administrative role workspace with the trusted path attribute and the script name must be in one of the account's execution profiles.

Most commands run during Trusted Solaris boot usually need to inherit privileges when starting, so entering the `/etc/init.d` pathname of a script to start a command would succeed only if the script is invoked in a shell that has the needed privileges and other attributes. And the `/etc/init.d/`*script* would have to be in one of the account's profiles with the needed privileges, unless the account has the Privileged Shells execution profile or has the `sysh` command in another execution profile.

See each command's man page for the privileges that a command needs. For example, the `sendmail(1MTSOL)` command's man page states that it needs the net_mac_read, net_privaddr, proc_nofloat, proc_setil, file_mac_read, and file_mac_search privileges to run the options that it runs with during boot and that it needs to run at ADMIN_HIGH with an effective UID of 0, so the command in the above example to start `sendmail` would work only if run with the specified attributes.

For example, the root role has the Privileged Shells profile, so root can enter `pfsh` followed by `sysh` to run the system shell with all privileges, before running either of the commands shown above.

# 16

## Security Administrator's Tasks in Adding Software

The default Trusted Solaris programs and actions have already been assigned privileges, effective UIDs or effective GIDs when any of these are required for the programs or actions to do their work. This section discusses the issues and tasks associated with the adding of the following types of software:

- Sun unbundled products and third-party applications that neither understand nor enforce Trusted Solaris security policy
- New programs, created by in-house developers 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)
- Shell scripts (created or approved by the security administrator)

Some programs run at a single level with no privileges required, so the security administrator can simply install them at ADMIN_LOW in a public directory and assign them as desired as commands in the execution 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, but before that is done, analysis and testing is required.

### Issues Around the Adding of Privileges to Any Software

To recap a basic principle of privilege management, software needs privileges or other security attributes only when it needs to circumvent some aspect of the system security policy in order to do its work.

To find out what privileges a program needs, the security administrator can turn on the `tsol_privs_debug` switch in `system`(4) and use `runpd` as described in "To Find Out Which Privileges an Application Needs" on page 473 to find out what privileges are being requested, but that is only the beginning.

If a program needs to override DAC or MAC restrictions on accessing a file, the security administrator might decide to assign an effective UID or GID to make the privilege unnecessary, as described in "Alternatives to Assigning Privilege" on page 440.

When software has been assigned privileges or an alternate UID or GID, the software becomes *trusted* by virtue of the fact that it is being allowed to bypass aspects of the Trusted Solaris security policy. The problem is that if you allow software to bypass security policy, you can make software *trusted* even though it might not be *trustworthy*. The security administrator must 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*.

## *When Adding Existing Programs*

Do the following when your site wishes to add any existing programs to a Trusted Solaris system, whether it is an application written outside of your organization, a Solaris unbundled software program, or a program written in house:

1. Find out if the application can run at all in the Trusted Solaris system.
   If it runs without privilege or any modification, you are done.

2. Find out why the program failed,
   Some unbundled software packages and third-party applications written for Solaris cannot run because of certain modifications made to the Trusted Solaris operating system 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 application is linked to the kernel or relies on aspects of the operating system that have been modified, the program probably is not capable of running on Trusted Solaris even if privileges were added unless other code changes were made.

3. If the program does not rely on aspects of the Solaris operating system that have been modified for Trusted Solaris, but it fails without privileges, find out what privileges or other attributes it needs.

4. If the program does require the use of privilege, assess whether the program will use its privileges in a trustworthy manner.
   See “Things to Think About When a Program Fails Without Privileges” on page 462.

5. If the program can safely run with the privileges or other attributes it requires in a manner that does not violate the Trusted Solaris security policy or the security policy of your installation, you may then assign the required privilege(s) as described in "How Privileges Are Assigned to Commands and Actions" on page 453.

6. If you have access to the source code of a program, you may add privileges in some cases, after a security consultant or programmer knowledgable about security modifies the code.
   These modifications might include privilege bracketing or adding code that makes the program aware of the Trusted Solaris security policy.

7. 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 "Why Privileged Programs Need to Use Trusted Shared Libraries" on page 454.

8. If the program cannot use its privileges in a trustworthy manner and it cannot be modified, do not make it available.

## *Things to Think About When a Program Fails Without Privileges*

The most obvious type of program that fails without privileges under Trusted Solaris is one that executes with setuid `root`. This kind of program may be assigned an effective UID of root in a profile.

Most applications are written in environments that do not have the security mechanisms needed by evaluated systems at B1 and above such as MAC and information labeling. For this reason, it is necessary that the person who assesses the program thoroughly understands security and thoroughly understands what the new program is trying to accomplish.

Be especially careful when allowing any program to violate Trusted Solaris policies such as MAC and information labeling, which have no analogs in standard UNIX systems. While UNIX applications that need to violate DAC often are implemented to make careful checks before doing so on a user's behalf, a standard UNIX application certainly does not make similar checks about MAC. If you give such a program a MAC-override privilege, you may unintentionally provide a way for users to override MAC arbitrarily.

Some of the security considerations you must assess 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 knows enough to check for DAC permissions on a file to make sure the user who executed the `rcp` command has permission to access the file. But `rcp` has no knowledge of MAC restrictions. If you gave it the `file_mac_read` or `file_mac_write` privilege, `rcp` would not have the built-in ability to do the same kind 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 enforced 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, 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 code to do the necessary MAC checks.

Some software may need privileges for reasons that are not obvious. 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 `kmem`(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, the offending policy overrides may be removed without impact on 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.

## *When Applications Need to Be Installed as Root*

Often the software that installs a particular application or package requires a *real* UID of root in order to succeed. Because the profile mechanism only allows the security administrator to assign an *effective* UID to an application, assigning a UID of root to the installation program in a profile would not fill the

requirement. The security administrator can enable this type of application to be installed by doing the steps in “To Set Up an Application to Run WIth a Real UID of Root.”

## When Applications Need to Run As Root

When an application has been written to run as root, the security administrator 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
  See “To Set Up An Application to Run With An Effective 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.
  See “To Find Out Which Privileges an Application Needs” on page 473.
- Assign the root role (temporarily or permanently) to the desired account(s) and assign the command to one of the account’s profiles.
  See“To Set Up an Application to Run WIth a Real UID of Root.”

## When Adding a New Trusted Program

Even though the program’s developer can determine which privileges to put into the program’s source code, if the security administrator does not assign a privilege that the program needs, then the program cannot use the privilege. Because both the security administrator and the developer must work together to add privileges to trusted programs, the use of privilege by new programs can be said to be another Trusted Solaris administrative task that is under two-person control. If it cannot override the security policy, the program may not do all the things you expect it to do, or it may not even run in the Trusted Solaris system.

### Developer’s Responsibilities

A developer who writes a program to be added to a Trusted Solaris system 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. Work with the security administrator to place shared libraries linked to the program in a trusted directory as described in "Why Privileged Programs Need to Use Trusted Shared Libraries" on page 454 and use the trusted directory location when compiling the program in Step 1 in "To Allow Trusted Programs to Link to Trusted Libraries" on page 478.

See the *Trusted Solaris Developer's Guide* for additional instructions on how to use privileges in programs.

### *Security Administrator's Responsibilities*

As the security administrator, you must ensure that a program that uses Trusted Solaris system calls and routines to work within security policy does not compromise the security of the Trusted Solaris system 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 `runpd` as described in "To Find Out Which Privileges an Application Needs" on page 473.
3. Scrutinize the source code to make sure it behaves in a trustworthy manner when using the privileges it needs to operate.

## *When Adding Actions*

The process of creating and using actions is pretty much the same in the Trusted Solaris system as it is in the base. Adding actions is described in the *CDE Advanced User's Guide and System Administrator's Guide.*

# 16

In Trusted Solaris, *use* of actions is controlled by the execution profile mechanism and by MAC. Actions may be assigned inheritable privileges in any execution profile, and they can run with privileges if they are invoked within one of the window system's trusted processes that can pass them privileges from its inheritable set. In the Trusted Solaris system, a number of actions have been assigned privileges in execution profiles that are assigned to certain roles by default. Table 16-3 summarizes the main differences encountered in creating and using actions in the Trusted Solaris system.

*Table 16-3* Differences in Creating and Using Actions Under Trusted Solaris Restraints *(1 of 2)*

| Base CDE | Trusted Solaris |
|---|---|
| New actions may be created by anyone within the originator's home directory, and a new action is automatically usable by its creator. | An action is only usable by a user or role if the action is one of the account's execution profiles. |
| | If the Create Action action or commands or actions that permit the editing of files are in an account's profile, the user or role *can* create a new action in the account's home directory, but the account may not be able to use the new action. |
| | There are two ways a user can use any new action: if the security administrator adds the name of the new action to one of the account's execution profiles, or if the person 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. |
| | Only if the account is allowed to use the action by its execution profiles, will the account be able to launch the action from its home directory through the File Manager. |

*Table 16-3* Differences in Creating and Using Actions Under Trusted Solaris Restraints *(2 of 2)*

| Base CDE | Trusted Solaris |
|---|---|
| A new action's files may be copied to public directories under `/etc/dt` to make the new actions available to others. | MAC restrictions prevent the copying of new actions to public directories. The public directories where actions are kept are not writable by normal users or non-administrative role accounts, so new actions cannot be moved to these public locations. |
| 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 `/usr/dt` and `/etc/dt` subdirectories where system-wide action files are kept. So, 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 cannot be 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 execution 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 accounts profiles. |

## *Modifying the Front Panel and Workspace Menu*

The *CDE Advanced User's Guide and System Administrator's Guide* describes how to modify components such as the Front Panel and Workspace Menu. Ordinary users are not allowed to modify the Front Panel and Workspace menu startup files, although they may drag and drop actions onto the Front Panel without security risks and therefore without restrictions.

Nobody but the security administrator can modify the configuration files for the Front Panel. The use of privileges by actions and their UID, GID, and sensitivity label are controlled through the profile mechanism, but any command specified in the configuration files for the Front Panel, its Subpanels, and the Workspace menu runs as root with all privileges. Because commands are not controlled but actions are, the security administrator must be cautious if doing customizations of any of these files and should use only `f.action` to launch an action, never use `f.exec` to execute a program.

For example, the `/usr/dt/config/C/sys.dtwmrc` file specifies the menus and the contents of the Front Panel. The example shows the menu line that launches Dtterm terminal.

```
"Terminal..."            f.action Dtterm
```

If the security administrator changed `f.action Dtterm` to `f.exec /usr/dt/bin/dtterm`, the terminal would come up as root with all privileges, while the Dtterm action would run with only the privileges specified for it in the account's profiles.

This manual contains two procedures that modify existing files to create new actions, one to create an alternate mail application that can run with privilege in the Front Panel and one to add an administrative action that can run with inherited privileges to the System_Admin folder for the purpose of editing another configuration file. See "To Create A New Administrative Action for Editing an Administrative File" on page 24 of Chapter 1 and "Substituting an Alternate Mail Application" on page 180 of Chapter 6.

Table 16-4 summarizes the main differences that apply to modifying the workspace under Trusted Solaris.

*Table 16-4* Differences in Modifying the Workspace Under Trusted Solaris Restraints

| Base CDE | Trusted Solaris |
|---|---|
| Any user can copy the system-wide version of the `dtwmrc`(4) file into $HOME`/.dt` and modify the workspace menu to bring up actions and invoke commands. | Even if a `dtwmrc` file is copied to any $HOME directory, the window system does not use that file. |
| Any action or command invoked through the workspace menu or from the Front Panel runs with the UID of the invoking user. | Any command invoked through the workspace menu or from the Front Panel runs as root with all privileges while any action invoked the same way would come up only with the privileges and other attributes specified for it in the account's profile. |

## *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 software and who is able to access a terminal or shell tool in which to run the shell script.

Making privileges available to commands run in shell scripts may be done only by security administrators. To begin with, here is a review of the Trusted Solaris constraints that affect how a shell script can be made to run with or without privileges.

Remember that the two ways any command can run with privilege are:

- The command's executable file has the needed privileges in both its forced and allowed sets, or
- The command has the needed privileges specified in a profile assigned to the person or role that invokes the command, the program file for the command has the needed privileges in its allowed set, and the command is being run in a profile shell from which it can inherit privilege

Making privileges available to a command within a shell script is not the same as making privileges available to a command when it is run on a command line, either in a shell tool or terminal emulator that is running a sh, csh, or ksh shell. Forced privilege commands can run with privilege in the sh, csh, and ksh shells because the forced privileges attached to the program file are available to the executing command even though the shell does not have any privileges. Assigning forced privileges to a sh, csh, or ksh shell script does not give any privileges to the commands executed by the shell script because even though the shell 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 "Process Privilege Sets" on page 445.

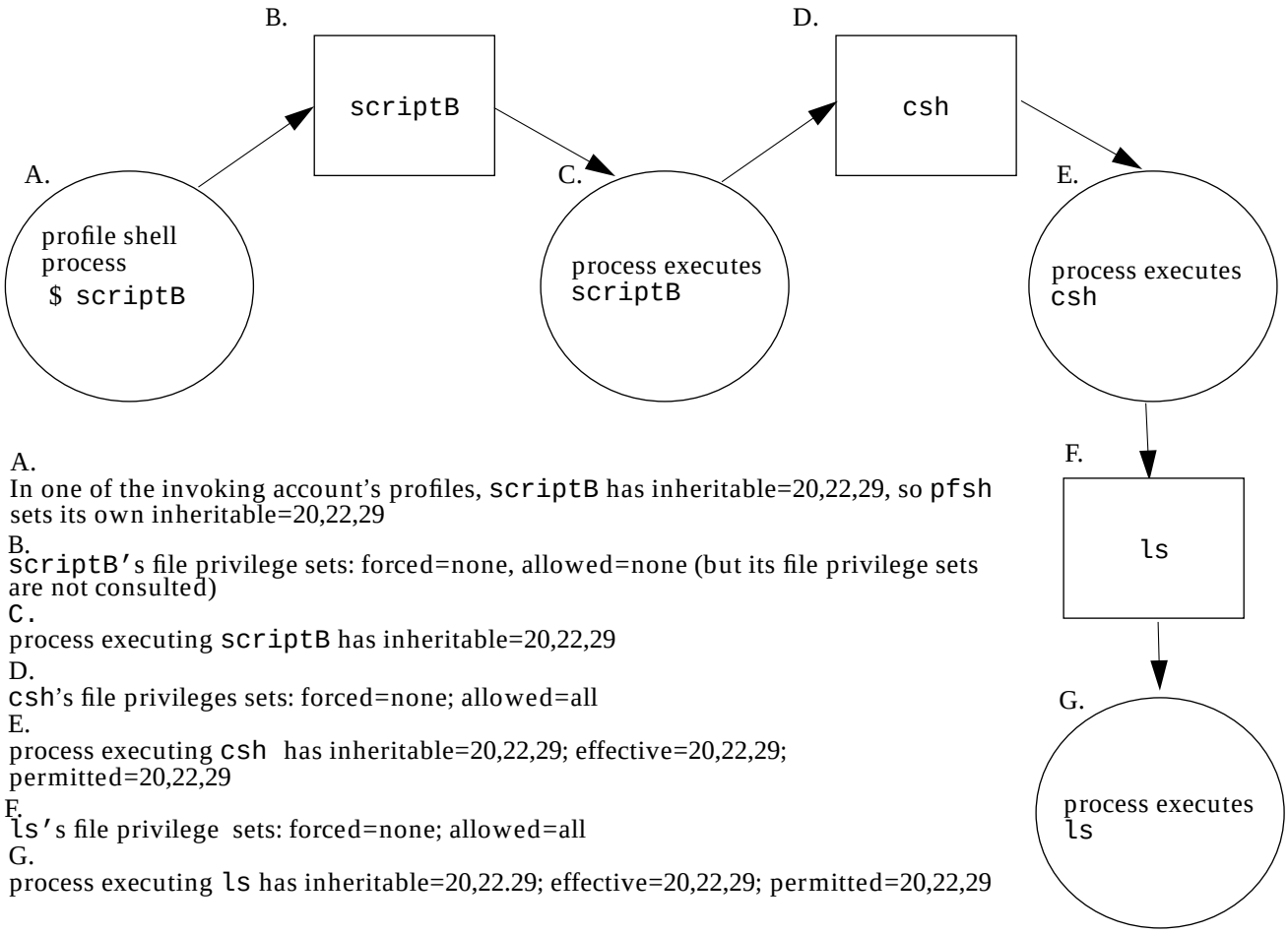
**Warning –** To prevent unauthorized tampering with object code or system scripts, whenever any executable program file is edited, any forced and allowed privileges previously given to that file are deleted. If a program file's allowed set is empty, it cannot use inheritable privileges, which are masked by the allowed set. However, in Trusted Solaris 2.5, the forced and allowed privileges for a shell script are not consulted by the profile mechanism when making inheritable privileges available. For this reason, shell scripts are more vulnerable than programs to being modified without detection. Before making shell scripts available that use inheritable privileges, the security administrator should keep in mind that the same protection against tampering that is available for programs is not available to shell scripts.

## *Summary of Shell Script Behavior in Trusted Solaris Systems*

- If none of its commands need privileges, a shell script can be created by anyone who is allowed to use a text editor and can used by anyone who has access to the software and who is able to access a terminal or shell tool in which to run the shell script.
- Forced privilege shell scripts do not pass privileges to commands that they contain.
- Allowed privileges on shell scripts have no effect on which privileges its programs can use.

  The allowed privilege set of the invoked shell's file is checked rather than that of the script's file.
- A standard shell script that is invoked in a profile shell can pass privileges to commands that it runs only if the shell script is listed with the privileges required by its commands in one of the invoking user's profiles.

  The shell script can pass any of its privileges to be inherited by the commands it executes if the commands themselves have the allowed privileges they need on their program files. 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—because standard shells do not consult the profiles database. See Figure 16-7 "How Shell Scripts Invoked in pfsh Can Pass Inheritable Privileges to Their Commands," on page 471.




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.
process executing `ls` has inheritable=20,22.29; effective=20,22,29; permitted=20,22,29

*Figure 16-7* How Shell Scripts Invoked in `pfsh` Can Pass Inheritable Privileges to Their Commands

See "To Write a Profile Shell Script That Runs Privileged Commands" on page 479.

- Shell scripts that use the profile shell can pass privileges to their commands in whatever shell they are running, if the *commands* are listed with the required privileges in one of the invoking user's profiles.

  See "To Write a Profile Shell Script That Runs Privileged Commands" on page 479.

### *How Program File Are Protected From Being Able to Use Inheritable Privileges If Edited*

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 can save the list of privileges on such a file before editing it and restore them afterwards, as described in "To Restore Privileges Lost When a File is Edited" on page 483.

## *Procedures for Adding Software*

### ▼ To Add A Package from a CD-ROM

1. **Assume the system administrator role, and in an ADMIN_LOW workspace, make a directory for the CD-ROM, using the File Manager New Folder option or `mkdir` on the command line.**

```
$ mkdir /cdrom
```

2. **Mount the CD-ROM.**

```
$ mount -o ro -F hsfs /dev/dsk/c0t6d0/s0 /cdrom
```

3. **Go to `/cdrom/Trusted_Solaris` and run `pkgadd` with the `-d` option followed by the name of the package.**

```
$ cd /cdrom/Trusted_Solaris
$ pkgadd -d . package_name
```

### ▼ To Set Up an Application to Run WIth a Real UID of Root

1. **Use the User Manager to assign the root role to the user account or administrative role account that is responsible for doing the installation.**
2. **Use the Profile Manager to assign the name of the application program to one of the profiles assigned to the root role.**
3. **If desired, use the User Manager to remove the root role from the account when the task is done.**

### ▼ To Set Up An Application to Run With An Effective UID of Root

1. **Use the Profile Manager to assign the command name of the application program to an appropriate profile and give the command a UID of root.**
2. **Use the User Manager to assign the profile with the added command to any user account or administrative role account as desired.**

### ▼ To Find Out Which Privileges an Application Needs

1. **As security administrator role in an ADMIN_LOW workspace, use the Admin Edit action to open the `/etc/system` file for editing.**
2. **Search for `set tsolsys` and set the value of `tsol_privs_debug=1`.** See Figure 16-8.

```
set tsolsys:tsol_privs_debug=1
```

*Figure 16-8* Changing the Value of `tsol_privs_debug` to `= 1` in the `system` File

3. **Save the changes to the file and close it.**
4. **Use the Admin Edit action to open the `/etc/syslog.conf` file for editing.**

5. **Find the lines shown in Figure 16-9.**

```
# Un-comment the following line to enable privilege debugging with
# the runpd command.  See runpd(1MTSOL).
#kern.debug;                                    /var/log/privdebug.log
```

*Figure 16-9* Commented Privilege Debugging Line in `/etc/syslog.conf`

6. **Remove the comment (#) at the beginning of the line that begins** `kern.debug`**, and add** `local0.debug` **after** `kern.debug`**.**
   See Figure 16-10.

```
# Un-comment the following line to enable privilege debugging with
# the runpd command.  See runpd(1MTSOL).
kern.debug;local0.debug                          /var/log/privdebug.log
```

*Figure 16-10* Privilege Debugging Line with `local0.debug` String Added in `/etc/syslog.conf`

---

**Note –** Uncommenting `kern.debug` enables privilege-debugging of an application's use of system calls. Adding the `local0.debug` string enables debugging of X window calls and is needed for privilege- debugging window applications.

---

7. **Save the changes to the file and close it.**
8. **Create the log file in** `/var/log`**.**
   a. **In an ADMIN_HIGH workspace, navigate to the** `/var/log` **directory using the File Manager.**
   b. **Use the New File option from the File Manager File menu and enter** `privdebug.log` **in the New File Name field.**
   c. **Change the owner of the** `privdebug.log` **file to root.**
9. **Reboot the machine.**

```
$ reboot
```

10. **Assume the secadmin role (or a role that has `runpd` in one of its profiles).**

11. **On the command line in a `pfsh` at the appropriate sensitivity label, enter the `runpd` command followed by the full pathname of the command whose use of privilege you want to check.**

**Note –** To test which privileges a program needs when run as a normal user, you should run `runpd` with a user ID of someone who is not in a role. To do this, you could create an action including `runpd` and assign the EUID of a normal user to that action, and then put the action in a profile for the security administrator role.

**Note –** The sensitivity label at which to invoke `runpd` depends on who will be using the application you are testing. While an administrative role might run the application at one of the administrative labels, a normal user would run the application at one of the labels in the user accreditation range.

As shown in Figure 16-11, `runpd` displays the name of the privilege(s) that the program needs in order to succeed followed by the type of access attempted (create in the example) followed by the name of the resource (RAW_SOCKET in the example).

```
$ ping redondo
ping: socket: Not owner
$ runpd /usr/sbin/ping redondo
redondo is alive
runpd: child terminated with a status of 0
process /usr/sbin/ping pid 1096 lacking privilege net_rawaccess
to perform create method upon resource RAW_SOCKET (Mar 29 11:39)
$
```

*Figure 16-11* `runpd` Displaying Privilege Needed For A Process To Succeed

12. **See also the log file for the privilege debugging messages.**
    A typical privilege debugging log entry looks like the example shown in Figure 16-12.

```
$ cat /var/log/privdebug.log
 Mar 29 12:18:43 crazy unix: DEBUG: /usr/sbin/ping pid 294 lacking
privilege 36 to 5 79
```

*Figure 16-12* Typical Privilege Debugging Entry in `/var/log/privedebug.log`

    The privilege numbers appear after the word "privilege." In the example above, the command `ping` is lacking privilege numbers 36. You can, if necessary, look up the 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.

13. **To give the needed privileges see “To Give Forced Privileges to a Command” on page 476 or and Chapter 8, “Managing Execution Profiles for Users and Roles” for how to use the Profile Manager to assign inheritable privileges.**

**Note –** For a command to be able to use either forced or allowed privileges, the privileges must be available in the command’s allowed privilege set.

14. **To turn off privilege debugging, restore all the changes you made to the `/etc/system` file and the `/etc/syslog.conf` file in Step 1 on page 473 through Step 7 on page 474 and reboot the machine.**

## ▼ To Give Forced Privileges to a Command

1. **As the file’s owner, as any user with the act as file owner authorization, or as the security administrator, go to the directory where the program file is located.**
   Use the File Manager to navigate to the directory or use `cd(1)` on the command line.

2. **Make sure the file is executable.**
   Use the File Manager—Permissions dialog box to make sure that the Execute box is checked for Owner, Group and Other. Alternately, if you have the `chown` command in your profiles and are the owner of the file, if you are in the security administrator role, or if you have the *change file owner* authorization, you may use `chown(1)` on the command line to make the command executable by everyone.

3. **Make sure the command has allowed privileges equal to the forced privileges you plan to assign.**

   a. **If you are using the File Manager—Permissions dialog box, select the Allowed button, assign Allowed Privileges, and then select the Forced button, and assign the Forced Privileges.**

   b. **If you have `setfpriv(1TSOL)` with the needed privileges in one of your profiles (as the security administrator role does in the default configuration), use `setfpriv` to assign the same privileges in both the allowed and forced sets.**
      The example shows the setting of file_dac_read and file_dac_write as allowed and forced privileges.

```
$ setfpriv -s -f file_dac_read,file_dac_write -p file_dac_read,file_dac_write test.priv.file
```

## ▼ To Allow Trusted Programs to Link to Trusted Libraries

1. **If adding a newly-created privileged application, do the following steps.**

   a. **Move the libraries that the application uses into one of the default trusted library directories in Table 16-5**

*Table 16-5* Default Directories for Trusted Shared Libraries

| | |
|---|---|
| **Trusted C function libraries** | `/usr/lib` and `/etc/lib` |
| **Trusted extensions to X Server** | `/usr/openwin/lib` and `/usr/dt/lib` |

For example for a privileged Java application that uses `/java/lib` libraries, move the `/java/lib` directory and the files it contains into `/usr/lib`, changing the `permissions` to 755.

```
$ ls /usr/lib
. . .
-r-xr-xr-x   1 bin          1912 Oct 25  1995 getoptcvt*
drwxr-xr-x   2 bin          1024 Dec  5  1995 iconv/
drwxr-xr-x   2 bin          1024 Dec  5  1995 java/
**. . .
```

   b. **Give the pathname of the trusted library to the programmer to use while linking the program.**

2. **If porting an existing application that needs privileges, add the directory name for the library to the `rtld` file.**

**Note –** Because the libraries for an existing application are fixed, you need to identify the library directory as trusted in the `/etc/security/tsol/rtld`.

   a. **Use the Admin Edit action to create or open the `/etc/security/tsol/rtld` file for editing.**

   b. **Specify the pathname of the library directory.**
   The example shows the path of a Java library directory used by a privileged Java application.

```
/usr/java/lib
```

   c. **Save and quit the `rtld` file.**

## ▼ To Write a Profile Shell Script That Runs Privileged Commands

1. **Start the script with `/bin/pfsh` (instead of another shell) on the first line.**

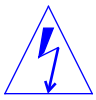
```
#!/bin/pfsh
```

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.

```
#!/bin/pfsh
# Copyright (c) 1993, 1997 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. **Use the Profile Manager to update an appropriate profile to list each of the commands that need to run within the shell script and to assign the commands the privileges they need.**
   To continue with the example, to enable root to run the example `cron` script with the needed privileges, the security administrator uses the Profile Manager to update the Object Label Management profile (which is assigned

to root by default) to include 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 privilege.



**Warning –** Be aware that 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 also whenever they are invoked on the command line in the profile shell—as long as the profile is in effect for the invoking account. So, for example, if you put `tfind` and `rm` into the Object Label Management profile with the privileges listed in the example, they then will execute in a profile shell with those privileges for whatever account has the Object Label Management execution profile as well as whenever the shell script is invoked by anyone in a standard shell.

### ▼ To Write a Standard Shell Script That Runs Privileged Commands When Executed in a Profile Shell

1. **Start the script with any standard shell (not `/bin/pfsh`) on the first line.**

```
#!/bin/csh
```

2. **Determine which commands need privileges and which privileges are needed.**
   The example, called `autosetpriv`, would allow 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 for the sake of example. A normal shell script would accept the privileges and the filename as arguments and do error checking. Do not use this shell script unless you want to assign the named privileges to an executable file called `executable`, which would then have those forced and allowed privileges available no matter who executes it.

```
#/bin/csh
setfpriv -s -f ipc_mac_write,ipc_upgrade_il,proc_setsl,sys_trans_label
-a ipc_mac_write,ipc_upgrade_il,proc_setsl,sys_trans_label executable
```

3. **Assume the security administrator role.**

4. **Use the Profile Manager to list the shell script in an appropriate profile and to give the script all of the privileges needed by any commands that run within the shell script.**
   To enable the script called `autosetpriv` to run with the file_setpriv privilege needed by the `setfpriv` command, the security administrator would use the Profile Manager to update the Object Privileges profile (which is assigned to secadmin 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.**

```
$ /bin/pfsh
$ autosetpriv
```

## ▼ To Specify Commands to Run With or Without Special Attributes During Boot

1. **Assume the security administrator role.**

2. **Create a new script to start and stop the desired command, using the system shell,** `sysh(1MTSOL)`**.**
   See “Run Control Scripts” in the Solaris *System Aministration Guide, Volume 1* for the basics and see the `sysh(1MTSOL)` man page.

```
#!/bin/sysh
```

3. **Put the new script into the `/etc/init.d` directory.**
   Usually the name given to the script is the name of the command it starts and stops. For example, the name of the script in `/etc/init.d` to start and stop `sendmail` is `sendmail`.

4. **For each run level at which the command should be started or stopped, go to the appropriate `/etc/rc`*n*`.d` directory and create a hard link from a properly-named target file to the `/etc/init.d` directory.**

   a. **Use the proper prefix in the target file's name for either starting (S) or stopping (K).**

   b. **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.**

```
$ cd /etc/rc2.d
$ ln /etc/init.d/scriptname [S|K]nnscriptname
```

   If the command does not need security attributes you are done.

5. **If the command started by the script needs any security attributes to succeed, bring up the Profile Manager and pick a naming service.**

   a. **If the command needs to start before networking services are up, select None from the Naming Service menu on the Profile Manager, so that the profile is created locally in `/etc/security/tsol/tsolprof`.**

   b. **If the command needs to start after networking services are up, select NIS+ from the Naming Service menu on the Profile Manager, so that the profile is created in the `tsolprof` NIS+ map on the NIS+ master.**

6. **Use the Profile Manager to create a new profile that lists the command and assigns privileges, UID, GID, or a sensitivity label, as needed.**
   For an example from the default Trusted Solaris system, the `sendmail` daemon is started by the `/etc/rc2.d/S88sendmail` script during boot with the `-bd` and `-q1h` arguments. The man page describes that `sendmail` needs to run with an effective UID of 0 at ADMIN_LOW and needs the net_mac_read, net_privaddr, proc_nofloat, proc_setil, file_mac_read, and file_mac_search privileges for the `-bd` and `-q` options, so the `sendmail` command is configured in the boot profile with UID of 0, a sensitivity label of ADMIN_LOW and with the required privileges.

7. **If the command started by the script needs any security attributes to succeed, use the `sysh`'s `setprof` option in the `/etc/init.d` shell script to identify the name of an execution profile.**

   a. **Use the Admin Edit action to bring up the script file in `/etc/init.d` for editing.**

   b. **Type in the `setprof` command followed by the name of the new profile created in Step 5.**

```
setprof new_profile_name
```

   c. **Save and close the script.**

8. **Reboot the system.**

```
$ reboot
```

### ▼ To Restore Privileges Lost When a File is Edited

1. **As security administrator, use `getfpriv(1TSOL)` to list the privileges on the file and direct the output into a temporary file.**

2. **After editing the program, use the File Manager to make the file executable (if needed) and to restore the privileges.**

# 16

# Host Administration Checklist 17

This chapter provides a checklist that you can photocopy and use as a worksheet when installing new hosts on the netwrok. The first column identifies the dialog box and the second column presents the information that needs to be supplied. The third column is provided for your convenience so that you can write down your intended input prior to running the software.

*Table 17-1* Host Administration Checklist *(1 of 2)*

| Dialog Box Title | Contents | Your Answer |
|---|---|---|
| Host Name | | |
| Networked? | Yes \| No | |
| IP address | | |
| Ethernet address | | |
| Primary network interface | Interfaces of workstation. | |
| Name service | NIS+ \| None | **None.** |

*Table 17-1* Host Administration Checklist *(2 of 2)*

| Dialog Box Title | Contents | Your Answer |
|---|---|---|
| Trusted Solaris configuration | Multiple user Sensitivity Labels?<br>Hide Upgraded Names?<br>Enable ILs?<br>Float ILs?<br>Reset ILs upon EXEC? | |
| Subnet | Yes \| No | |
| Subnet mask | 255.255.255.0 | |
| Time zone | Offset from GMT \| Geographical | |
| Date and Time | | |
| System type | Standalone \| OS server \| Diskless | |
| Software group | End user \| Developer \| Entire | |
| Customize? | Yes \| No | |
| Disk(s) to use | Disks visible to the workstation. | |
| Preserve? | Disks to leave as they are. | |
| Auto-layout file systems? | Yes \| No | |
| File systems to auto-layout | `/, /usr, /var, /opt, swap` | |
| Begin installation | Yes \| No | |
| Reboot | Yes \| No | |
| Root password | | |

# *Profile Summary Tables* A

This appendix provides three tables for working with execution profiles. Your site may have customized the default profiles, and if so, the customizations will not be reflected here.


| | |
|---|---|
| *Execution Profile Content Summary* | *page 488* |
| *Finding Commands in Execution Profiles* | *page 493* |
| *Finding Actions in Execution Profiles* | *page 506* |

# A

## *Execution Profile Content Summary*

Table A-1 lists each execution profile and the commands, actions, and authorizations assigned to it in the default configuration. The table also indicates in parentheses () to which roles each profile is assigned by default. If you need the specific security attributes assigned to actions and command in a particular profile, see the individual profile tables in Appendix B, “Profile Definition Tables” or use the Profile Manager to view that profile.

*Table A-1* Execution Profile Contents

| **Profile (Default Role Assignments)** | | |
|---|---|---|
| **Commands** | **Actions** | **Authorizations** |
| **All (Root)** | | |
| All commands | All actions | |
| **All Authorizations (Root)** | | |
| | | All authorizations |
| **Audit Control Profile (Security Administrator)** | | |
| bsmconv, bsmunconv, mkdir, writeaudit, audit, auditconfig, auditd, auditstat, format, mkfs, mount, mountall, newfs, newsecfs, share, shareall, tunefs, umount, umountall, unshare, unshareall | AuditClass, AuditControl, AuditEvent, AuditStartup, AuditUser | set user audit flags, set / get file audit flags, act as file owner |
| **Audit Review Profile (System Administrator)** | | |
| awk, cat, grep, sed, tail, auditreduce, praudit | | |
| **Basic Actions (Security Administrator, System Administrator, System Operator, Root)** | | |
| tsolxagent, ttsession | Dtcalc, Dtcm, Dtfile, Dthelpview, SDTimage, Dtmail, Dtmanpageview, Dtterm, Dtpad, Dttrash, Print, Dtappmgr, DtTTMediaOpen, DtfileHome, InvokeFILEMGR, InvokeMAILER, OpenFolder, DtUnlink, Open, TextEditor, Trash | |

*Table A-1* Execution Profile Contents

| **Profile (Default Role Assignments)** | | |
|---|---|---|
| **Commands** | **Actions** | **Authorizations** |
| **Basic Commands (Security Administrator, System Administrator, System Operator, Root)** | | |
| adminvi, awk, cat, cd, chmod, clear, cmp, col, compress, cp, cut, df, diff, diff3, dircmp, dirname, du, echo, env, expr, false, fgrep, file, fold, getlabel, grep, head, hostid, hostname, id, join, ldd, ln, look, lp, ls, mailq, man, mkdir, more, mv, niscat, nisdefaults, niserror, nisgrep, nismatch, nistest, nroff, page, pfsh, pg, pr, pwd, rcp, rlogin, rm, rmdir, script, sdiff, sleep, sort, spell, stty, tail, tbl, test, tfind, time, touch, troff, true, tty, uname, uncompress, uniq, which, who, rsh, whereis, whoami | | |
| **Convenient Authorizations** | | |
| | | enable logins, modify cron admin, print a PostScript file, print without labels, remote login, set application search path |
| **Enable Login (System Administrator)** | | |
| | | enable logins |
| **Maintenance and Repair (System Administrator)** | | |
| autopush, clri, adb, netstat, add_drv, aspppd, cachefslog, clri, crash, eeprom, fsck, fsdb, fsirand, grpck, halt, ncheck, ping, poweroff, prtconf, pwck, reboot, rem_drv, snoop, spray, strace, syslogd, tokmapd, tunefs | | enable logins, remote login, terminal login |
| **Media Backup(System Operator)** | | |
| mt, tar, ufsdump | Tar, TarList, OWtapetool, TarList, TarUnpack, OWtapetool | allocate device |
| **Media Restore (System Administrator)** | | |
| cpio, mt, tar, ufsrestore | TarList, TarUnpack, OWtapetool | allocate device |

# A

*Table A-1* Execution Profile Contents

| Commands | Actions | Authorizations |
|---|---|---|
| **Profile (Default Role Assignments)** | | |
| **NIS+ Administration (System Administrator)** | | |
| nischttl, nisln, nisctl, nisping, nisshowcache, nisstat, nistnsetup, nistntime, nscd | | |
| **NIS+ Security Administration (Security Administrator)** | | |
| chkey, nisaddcred, nischgrp, nischmod, nischown, nisgrpadm, nismkdir, nispasswd, nisrm, nisrmdir, nistbladm, nisaddent, nisclient, nispopulate, nisserver, nissetup, nisupdkeys, newkey, nisinit, nislog | | |
| **Object Access Management (Security Administrator)** | | |
| chgrp, chmod, chown, getfacl, getfattrflag, getlabel, setfacl, setfattrflag | Dtfile, Dttrash, DtfileHome, InvokeFILEMGR | act as file owner, change file owner |
| **Object Label Management (Security Administrator)** | | |
| setfattrflag, cp, getlabel, getmldadorn, getsldname, mldpwd, mldrealpath, mv, rm, setfattrflag, setlabel, tfind, procplabel, atohexlabel, chk_encodings, hextoalabel, tokmapctl | Dtfile, Dttrash, CheckEncodings, EditEncodings, DtfileHome | allocate device, bypass view of file contents on drag and drop, downgrade file sensitivity label, upgrade file sensitivity label, use all defined labels |
| **Object Privilege Management** | | |
| adb, getfpriv, kill, ldd, pkginfo, setfpriv, testfpriv, truss, procppriv, procpprivtest, pkgadd, pkgchk, pkgrm, runpd, swmtool | Dtfile, DtfileHome, InvokeFILEMGR | set file privileges |
| **Outside Accred (Security Administrator, System Administrator, System Operator)** | | |
| | | use all defined label |
| **Privileged Shells (Root)** | | |
| csh, ksh, sh | | |

*Table A-1* Execution Profile Contents

| Commands | Actions | Authorizations |
|---|---|---|
| **Profile (Default Role Assignments)** | | |
| **System Management (System Administrator)** | | |
| adminvi, cancel, date, disable, enable, kill, lp, lpstat, mailq, newaliases, nice, ps, rdist, renice, rup, vmstat, xhost, pattr, pclear, pcred, pfiles, pflags, plabel, pldd, pmap, prun, psig, pstack, pstop, ptime, ptree, pwait, pwdx, accept, allocate, deallocate, format, getfsattr, in.named, init, list_devices, lpadmin, lpfilter, lpmove, lpshut, lpsystem, lpusers, mkfile, mkfs, mount, mountall, newfs, ping, reject, share, shareall, showmount, swap, umount, umountall, unshare, unshareall | Dbmgr, Hostmgr, DNS_Resolve, EditMotd, Vfstab, ShareFS | administer printing, modify at users, modify bootparams, modify cron users, modify ethers, modify hosts, modify locale, modify netmasks, modify networks, modify protocols, modify rpc, modify services, modify timezone |
| **System Security (Security Administrator)** | | |
| adminvi, disable, enable, ps, accept, add_drv, allocate, autopush, deallocate, drvconfig, format, getfsattr, mkfs, newfs, newsecfs, ping, reject, rem_drv, setfsattr, tnchkdb, tnctl, tnd, tninfo | Dbmgr, Printermgr, Serialmgr, TrustedEditor, Nsswitch, SendMail, Vfstab_adjunct | modify at admin, modify cron admin, modify netgroup, modify tnidb, modify tnrhdb, modify tnrhtp, print a PostScript file, print without labels, remote login |
| **User Management (System Administrator)** | | |
| groupmgr, usermgr | Groupmgr, Usermgr | modify aliases, set attributes related to home directories, set user identity |
| **User Security (Security Administrator, Root)** | | |
| dbmgr, profmgr, usermgr | | modify auto_home, set idle time, set list of assumable roles, set user audit flags, set user labels, set user password, set user profiles, set/get file audit flags, use all defined labels |

# A

*Table A-1* Execution Profile Contents

| **Profile (Default Role Assignments)** | | |
|---|---|---|
| **Commands** | **Actions** | **Authorizations** |
| **boot** | | |
| mountall, lpsched, auditd, cron, in.named, inetd, nscd, rpcbind, syslogd, tnd | | act as file owner, allocate device, bypass view of file contents on drag and drop, change file owner, downgrade file sensitivity label, enable logins, paste to a downgraded window, paste to an upgraded window, permit self-modification, print a PostScript file, print without labels, remote login, set application search path, set attributes related to home directories, set file privileges, set idle time, set list of assumable roles, set user audit flags, set user identity, set user labels, set user password, set user profiles, set/get file audit flags, terminal login, upgrade file sensitivity label, use all defined labels |
| **inetd** | | |
| rpc.cmsd, rpc.ttdbserverd, in.ftpd, in.rexecd, in.rlogind, in.rshd, in.telnetd, in.tftpd, sadmind | | |

## *Finding Commands in Execution Profiles*

Table A-2 lists each command contained in any execution profile and the execution profile(s) to which it is assigned. Remember that a command may be listed in more than one execution profile and could have a different set of security attributes in each profile in which it is listed. The table also indicates the full path of the command as well as whether it has any security attributes: minimum label, maximum label, setUID value, setGID value, and privileges. The term *privs* indicates that the command has one or more privileges. For specific privilege information, see the individual profile tables in Appendix B, "Profile Definition Tables" or use the Profile Manager to determine which privileges are applied to the command within that profile.

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| accept | System Management | /usr/sbin/accept | privs |
| accept | System Security | /usr/sbin/accept | privs |
| adb | Maintenance and Repair | /usr/bin/adb | |
| adb | Object Privilege Management | /usr/bin/adb | |
| add_drv | Maintenance and Repair | /usr/sbin/add_drv | |
| add_drv | System Security | /usr/sbin/add_drv | |
| adminvi | Basic Commands | /usr/bin/adminvi | |
| adminvi | System Management | /usr/bin/adminvi | |
| adminvi | System Security | /usr/bin/adminvi | |
| allocate | System Management | /usr/sbin/allocate | privs |
| allocate | System Security | /usr/sbin/allocate | privs |
| aspppd | Maintenance and Repair | /usr/sbin/aspppd | |
| atohexlabel | Object Label Management | /usr/sbin/atohexlabel | |
| audit | Audit Control | /usr/sbin/audit | UID = 0, privs |
| auditconfig | Audit Control | /usr/sbin/auditconfig | min label = ADMIN_LOW, max label = ADMIN_LOW, UID = 0, privs |
| auditd | Audit Control | /usr/sbin/auditd | UID = 0, privs |
| auditd | boot | /usr/sbin/auditd | privs |

# A

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| auditreduce | Audit Review | /usr/sbin/auditreduce | min label = ADMIN_HIGH, UID = 0, privs |
| auditstat | Audit Control | /usr/sbin/auditstat | UID = 0 |
| autopush | Maintenance and Repair | /etc/autopush | |
| autopush | System Security | /usr/sbin/autopush | |
| awk | Audit Review | /usr/bin/awk | min label = ADMIN_HIGH, max label = ADMIN_HIGH, UID = 0 |
| awk | Basic Commands | /usr/bin/awk | |
| bsmconv | Audit Control | /etc/security/bsmconv | min label = ADMIN_LOW, max label = ADMIN_LOW, UID = 0 |
| bsmunconv | Audit Control | /etc/security/bsmunconv | min label = ADMIN_LOW, max label = ADMIN_LOW, UID = 0 |
| cachefslog | Maintenance and Repair | /usr/sbin/cachefslog | |
| cancel | System Management | /usr/bin/cancel | UID = 71 |
| cat | Audit Review | /usr/bin/cat | min label = ADMIN_HIGH, max label = ADMIN_HIGH, UID = 0 |
| cat | Basic Commands | /usr/bin/cat | |
| cd | Basic Commands | /usr/bin/cd | |
| chgrp | Object Access Management | /usr/bin/chgrp | privs |
| chkey | NIS+ Security Administration | /usr/bin/chkey | |
| chk_encodings | Object Label Management | /usr/sbin/chk_encodings | |
| chmod | Basic Commands | /usr/bin/chmod | |
| chmod | Object Access Management | /usr/bin/chmod | privs |
| chown | Object Access Management | /usr/bin/chown | privs |
| clear | Basic Commands | /usr/bin/clear | |
| clri | Maintenance and Repair | /etc/clri | |

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| clri | Maintenance and Repair | /usr/sbin/clri | |
| cmp | Basic Commands | /usr/bin/cmp | |
| col | Basic Commands | /usr/bin/col | |
| compress | Basic Commands | /usr/bin/compress | |
| cp | Basic Commands | /usr/bin/cp | |
| cp | Object Label Management | /usr/bin/cp | privs |
| cpio | Media Restore | /usr/bin/cpio | |
| crash | Maintenance and Repair | /usr/sbin/crash | |
| cron | boot | /usr/sbin/cron | min label = ADMIN_LOW, max label = ADMIN_HIGH, UID = 0, privs |
| csh | Privileged Shells | /usr/bin/csh | privs |
| cut | Basic Commands | /usr/bin/cut | |
| date | System Management | /usr/bin/date | privs |
| dbmgr | User Security | /opt/SUNWadm/2.1/bin/dbmgr | privs |
| deallocate | System Management | /usr/sbin/deallocate | privs |
| deallocate | System Security | /usr/sbin/deallocate | privs |
| df | Basic Commands | /usr/bin/df | |
| diff | Basic Commands | /usr/bin/diff | |
| diff3 | Basic Commands | /usr/bin/diff3 | |
| dircmp | Basic Commands | /usr/bin/dircmp | |
| dirname | Basic Commands | /usr/bin/dirname | |
| disable | System Management | /usr/bin/disable | privs |
| disable | System Security | /usr/bin/disable | privs |
| drvconfig | System Security | /usr/sbin/drvconfig | |
| du | Basic Commands | /usr/bin/du | |
| echo | Basic Commands | /usr/bin/echo | |
| eeprom | Maintenance and Repair | /usr/sbin/eeprom | |

# A

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| enable | System Management | /usr/bin/enable | privs |
| enable | System Security | /usr/bin/enable | privs |
| env | Basic Commands | /usr/bin/env | |
| expr | Basic Commands | /usr/bin/expr | |
| false | Basic Commands | /usr/bin/false | |
| fgrep | Basic Commands | /usr/bin/fgrep | |
| file | Basic Commands | /usr/bin/file | |
| fold | Basic Commands | /usr/bin/fold | |
| format | Audit Control | /usr/sbin/format | UID = 0, privs |
| format | System Management | /usr/sbin/format | UID = 0, privs |
| format | System Security | /usr/sbin/format | UID = 0, privs |
| fsck | Maintenance and Repair | /usr/sbin/fsck | |
| fsdb | Maintenance and Repair | /usr/sbin/fsdb | |
| fsirand | Maintenance and Repair | /usr/sbin/fsirand | |
| getfacl | Object Access Management | /usr/bin/getfacl | privs |
| getfattrflag | Object Access Management | /usr/bin/getfattrflag | privs |
| getfpriv | Object Privilege Management | /usr/bin/getfpriv | |
| getfsattr | System Management | /usr/sbin/getfsattr | privs |
| getfsattr | System Security | /usr/sbin/getfsattr | GID = 3 |
| getlabel | Basic Commands | /usr/bin/getlabel | |
| getlabel | Object Access Management | /usr/bin/getlabel | privs |
| getlabel | Object Label Management | /usr/bin/getlabel | privs |
| getmldadorn | Object Label Management | /usr/bin/getmldadorn | |
| getsldname | Object Label Management | /usr/bin/getsldname | |
| grep | Audit Review | /usr/bin/grep | min label = ADMIN_HIGH, max label = ADMIN_HIGH, UID = 0 |
| grep | Basic Commands | /usr/bin/grep | |

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| groupmgr | User Management | /opt/SUNWadm/2.1/bin/groupmgr | |
| grpck | Maintenance and Repair | /usr/sbin/grpck | |
| halt | Maintenance and Repair | /usr/sbin/halt | UID = 0, privs |
| head | Basic Commands | /usr/bin/head | |
| hextoalabel | Object Label Management | /usr/sbin/hextoalabel | privs |
| hostid | Basic Commands | /usr/bin/hostid | |
| hostname | Basic Commands | /usr/bin/hostname | |
| id | Basic Commands | /usr/bin/id | |
| in.ftpd | inetd | /usr/sbin/in.ftpd | privs |
| in.named | boot | /usr/sbin/in.named | min label = ADMIN_LOW, max label = ADMIN_HIGH, UID = 0, privs |
| in.named | System Management | /usr/sbin/in.named | min label = ADMIN_LOW, max label = ADMIN_HIGH, UID = 0, privs |
| in.rexecd | inetd | /usr/sbin/in.rexecd | privs |
| in.rlogind | inetd | /usr/sbin/in.rlogind | privs |
| in.rshd | inetd | /usr/sbin/in.rshd | privs |
| in.telnetd | inetd | /usr/sbin/in.telnetd | privs |
| in.tftpd | inetd | /usr/sbin/in.tftpd | privs |
| inetd | boot | /usr/sbin/inetd | min label = ADMIN_LOW, max label = ADMIN_HIGH, privs |
| init | System Management | /usr/sbin/init | privs |
| join | Basic Commands | /usr/bin/join | |
| kill | Object Privilege Management | /usr/bin/kill | |
| kill | System Management | /usr/bin/kill | privs |
| ksh | Privileged Shells | /usr/bin/ksh | privs |
| ldd | Basic Commands | /usr/bin/ldd | |

# A

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| ldd | Object Privilege Management | /usr/bin/ldd | |
| list_devices | System Management | /usr/sbin/list_devices | |
| ln | Basic Commands | /usr/bin/ln | |
| look | Basic Commands | /usr/bin/look | |
| lp | Basic Commands | /usr/bin/lp | |
| lp | System Management | /usr/bin/lp | |
| lpadmin | System Management | /usr/sbin/lpadmin | max label = ADMIN_LOW, UID = 0 |
| lpfilter | System Management | /usr/sbin/lpfilter | max label = ADMIN_LOW, UID = 0 |
| lpmove | System Management | /usr/sbin/lpmove | max label = ADMIN_LOW, UID = 0 |
| lpsched | boot | /usr/lib/lp/lpsched | min label = ADMIN_HIGH, max label = ADMIN_HIGH, privs |
| lpshut | System Management | /usr/sbin/lpshut | UID = 0 |
| lpstat | System Management | /usr/bin/lpstat | max label = ADMIN_LOW |
| lpsystem | System Management | /usr/sbin/lpsystem | max label = ADMIN_LOW, UID = 0 |
| lpusers | System Management | /usr/sbin/lpusers | max label = ADMIN_LOW, UID = 0 |
| ls | Basic Commands | /usr/bin/ls | |
| mailq | Basic Commands | /usr/bin/mailq | GID = 2 |
| mailq | System Management | /usr/bin/mailq | GID = 2, privs |
| man | Basic Commands | /usr/bin/man | |
| mkdir | Audit Control | /usr/bin/mkdir | privs |
| mkdir | Basic Commands | /usr/bin/mkdir | |
| mkfile | System Management | /usr/sbin/mkfile | |
| mkfs | Audit Control | /usr/sbin/mkfs | privs |
| mkfs | System Management | /usr/sbin/mkfs | UID = 0, privs |

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| mkfs | System Security | /usr/sbin/mkfs | UID = 0, privs |
| mldpwd | Object Label Management | /usr/bin/mldpwd | |
| mldrealpath | Object Label Management | /usr/bin/mldrealpath | |
| more | Basic Commands | /usr/bin/more | |
| mount | Audit Control | /usr/sbin/mount | UID = 0, privs |
| mount | System Management | /usr/sbin/mount | UID = 0, privs |
| mountall | Audit Control | /usr/sbin/mountall | UID = 0, privs |
| mountall | boot | /sbin/mountall | max label = ADMIN_HIGH, privs |
| mountall | System Management | /usr/sbin/mountall | UID = 0, privs |
| mt | Media Backup | /usr/bin/mt | |
| mt | Media Restore | /usr/bin/mt | |
| mv | Basic Commands | /usr/bin/mv | |
| mv | Object Label Management | /usr/bin/mv | privs |
| ncheck | Maintenance and Repair | /usr/sbin/ncheck | |
| netstat | Maintenance and Repair | /usr/bin/netstat | |
| newaliases | System Management | /usr/bin/newaliases | UID = 0, privs |
| newfs | Audit Control | /usr/sbin/newfs | UID = 0, privs |
| newfs | System Management | /usr/sbin/newfs | UID = 0, privs |
| newfs | System Security | /usr/sbin/newfs | UID = 0, privs |
| newkey | NIS+ Security Administration | /usr/sbin/newkey | |
| newsecfs | Audit Control | /usr/sbin/newsecfs | privs |
| newsecfs | System Security | /usr/sbin/newsecfs | UID = 0, privs |
| nice | System Management | /usr/bin/nice | |
| nisaddcred | NIS+ Security Administration | /usr/bin/nisaddcred | |
| nisaddent | NIS+ Security Administration | /usr/lib/nis/nisaddent | |
| niscat | Basic Commands | /usr/bin/niscat | |
| nischgrp | NIS+ Security Administration | /usr/bin/nischgrp | |

# A

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| nischmod | NIS+ Security Administration | /usr/bin/nischmod | |
| nischown | NIS+ Security Administration | /usr/bin/nischown | |
| nischttl | NIS+ Administration | /usr/bin/nischttl | |
| nisclient | NIS+ Security Administration | /usr/lib/nis/nisclient | privs |
| nisctl | NIS+ Administration | /usr/lib/nis/nisctl | |
| nisdefaults | Basic Commands | /usr/bin/nisdefaults | |
| niserror | Basic Commands | /usr/bin/niserror | |
| nisgrep | Basic Commands | /usr/bin/nisgrep | |
| nisgrpadm | NIS+ Security Administration | /usr/bin/nisgrpadm | |
| nisinit | NIS+ Security Administration | /usr/sbin/nisinit | |
| nisln | NIS+ Administration | /usr/bin/nisln | |
| nislog | NIS+ Security Administration | /usr/sbin/nislog | |
| nismatch | Basic Commands | /usr/bin/nismatch | |
| nismkdir | NIS+ Security Administration | /usr/bin/nismkdir | |
| nispasswd | NIS+ Security Administration | /usr/bin/nispasswd | |
| nisping | NIS+ Administration | /usr/lib/nis/nisping | |
| nispopulate | NIS+ Security Administration | /usr/lib/nis/nispopulate | |
| nisrm | NIS+ Security Administration | /usr/bin/nisrm | |
| nisrmdir | NIS+ Security Administration | /usr/bin/nisrmdir | |
| nisserver | NIS+ Security Administration | /usr/lib/nis/nisserver | privs |
| nissetup | NIS+ Security Administration | /usr/lib/nis/nissetup | |
| nisshowcache | NIS+ Administration | /usr/lib/nis/nisshowcache | |
| nisstat | NIS+ Administration | /usr/lib/nis/nisstat | |
| nistbladm | NIS+ Security Administration | /usr/bin/nistbladm | |
| nistest | Basic Commands | /usr/bin/nistest | |
| nistnsetup | NIS+ Administration | /usr/lib/nis/nistnsetup | |
| nistntime | NIS+ Administration | /usr/lib/nis/nistntime | |

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| nisupdkeys | NIS+ Security Administration | /usr/lib/nis/nisupdkeys | |
| nroff | Basic Commands | /usr/bin/nroff | |
| nscd | boot | /usr/sbin/nscd | min label = ADMIN_LOW, max label = ADMIN_HIGH, privs |
| nscd | NIS+ Administration | /usr/sbin/nscd | privs |
| page | Basic Commands | /usr/bin/page | |
| pattr | System Management | /usr/proc/bin/pattr | privs |
| pclear | System Management | /usr/proc/bin/pclear | privs |
| pcred | System Management | /usr/proc/bin/pcred | privs |
| pfiles | System Management | /usr/proc/bin/pfiles | privs |
| pflags | System Management | /usr/proc/bin/pflags | privs |
| pfsh | Basic Commands | /usr/bin/pfsh | |
| pg | Basic Commands | /usr/bin/pg | |
| ping | Maintenance and Repair | /usr/sbin/ping | |
| ping | System Management | /usr/sbin/ping | |
| ping | System Security | /usr/sbin/ping | |
| pkgadd | Object Privilege Management | /usr/sbin/pkgadd | |
| pkgchk | Object Privilege Management | /usr/sbin/pkgchk | |
| pkginfo | Object Privilege Management | /usr/bin/pkginfo | |
| pkgrm | Object Privilege Management | /usr/sbin/pkgrm | |
| plabel | System Management | /usr/proc/bin/plabel | privs |
| pldd | System Management | /usr/proc/bin/pldd | privs |
| pmap | System Management | /usr/proc/bin/pmap | privs |
| poweroff | Maintenance and Repair | /usr/sbin/poweroff | UID = 0, privs |
| pr | Basic Commands | /usr/bin/pr | |
| praudit | Audit Review | /usr/sbin/praudit | min label = ADMIN_HIGH, UID = 0, privs |

# A

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| procplabel | Object Label Management | /usr/proc/bin/plabel | privs |
| procppriv | Object Privilege Management | /usr/proc/bin/ppriv | privs |
| procpprivtest | Object Privilege Management | /usr/proc/bin/pprivtest | privs |
| profmgr | User Security | /opt/SUNWadm/2.1/bin/profmgr | privs |
| prtconf | Maintenance and Repair | /usr/sbin/prtconf | |
| prun | System Management | /usr/proc/bin/prun | privs |
| ps | System Management | /usr/bin/ps | privs |
| ps | System Security | /usr/bin/ps | privs |
| psig | System Management | /usr/proc/bin/psig | privs |
| pstack | System Management | /usr/proc/bin/pstack | privs |
| pstp | System Management | /usr/proc/bin/pstop | privs |
| ptime | System Management | /usr/proc/bin/ptime | privs |
| ptree | System Management | /usr/proc/bin/ptree | privs |
| pwait | System Management | /usr/proc/bin/pwait | privs |
| pwck | Maintenance and Repair | /usr/sbin/pwck | |
| pwd | Basic Commands | /usr/bin/pwd | |
| pwdx | System Management | /usr/proc/bin/pwdx | privs |
| rcp | Basic Commands | /usr/bin/rcp | |
| rdist | System Management | /usr/bin/rdist | |
| reboot | Maintenance and Repair | /usr/sbin/reboot | UID = 0, privs |
| reject | System Management | /usr/sbin/reject | privs |
| reject | System Security | /usr/sbin/reject | privs |
| rem_drv | Maintenance and Repair | /usr/sbin/rem_drv | |
| rem_drv | System Security | /usr/sbin/rem_drv | privs |
| renice | System Management | /usr/bin/renice | privs |
| rlogin | Basic Commands | /usr/bin/rlogin | |
| rm | Basic Commands | /usr/bin/rm | |

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| rm | Object Label Management | /usr/bin/rm | privs |
| rmdir | Basic Commands | /usr/bin/rmdir | |
| rpc.cmsd | inetd | /usr/dt/bin/rpc.cmsd | privs |
| rpc.ttdbserverd | inetd | /usr/dt/bin/rpc.ttdbserverd | privs |
| rpcbind | boot | /usr/sbin/rpcbind | min label = ADMIN_HIGH, max label = ADMIN_HIGH, privs |
| rsh | Basic Commands | /usr/ucb/rsh | |
| runpd | Object Privilege Management | /usr/sbin/runpd | |
| rup | System Management | /usr/bin/rup | |
| sadmind | inetd | /usr/sbin/sadmind | privs |
| script | Basic Commands | /usr/bin/script | |
| sdiff | Basic Commands | /usr/bin/sdiff | |
| sed | Audit Review | /usr/bin/sed | min label = ADMIN_HIGH, max label = ADMIN_HIGH, UID = 0 |
| setfacl | Object Access Management | /usr/bin/setfacl | privs |
| setfattrflag | Object Access Management | /usr/bin/setfattrflag | privs |
| setfattrflag | Object Label Management | /usr/bin/setfattrflag | |
| setfpriv | Object Privilege Management | /usr/bin/setfpriv | privs |
| setfsattr | System Security | /usr/sbin/setfsattr | privs |
| setlabel | Object Label Management | /usr/bin/setlabel | privs |
| sh | Privileged Shells | /usr/bin/sh | privs |
| share | Audit Control | /usr/sbin/share | UID = 0, privs |
| share | System Management | /usr/sbin/share | UID = 0, privs |
| shareall | Audit Control | /usr/sbin/shareall | UID = 0, privs |
| shareall | System Management | /usr/sbin/shareall | UID = 0, privs |
| showmount | System Management | /usr/sbin/showmount | |
| sleep | Basic Commands | /usr/bin/sleep | |

# A

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| snoop | Maintenance and Repair | /usr/sbin/snoop | |
| sort | Basic Commands | /usr/bin/sort | |
| spell | Basic Commands | /usr/bin/spell | |
| spray | Maintenance and Repair | /usr/sbin/spray | |
| strace | Maintenance and Repair | /usr/sbin/strace | |
| stty | Basic Commands | /usr/bin/stty | |
| swap | System Management | /usr/sbin/swap | privs |
| swmtool | Object Privilege Management | /usr/sbin/swmtool | |
| syslogd | boot | /usr/sbin/syslogd | min label = ADMIN_LOW, max label = ADMIN_HIGH, privs |
| syslogd | Maintenance and Repair | /usr/sbin/syslogd | |
| tail | Audit Review | /usr/bin/tail | min label = ADMIN_HIGH, max label = ADMIN_HIGH, UID = 0 |
| tail | Basic Commands | /usr/bin/tail | |
| tar | Media Backup | /usr/bin/tar | privs |
| tar | Media Restore | /usr/bin/tar | privs |
| tbl | Basic Commands | /usr/bin/tbl | |
| test | Basic Commands | /usr/bin/test | |
| testfpriv | Object Privilege Management | /usr/bin/testfpriv | privs |
| tfind | Basic Commands | /usr/bin/tfind | |
| tfind | Object Label Management | /usr/bin/tfind | privs |
| time | Basic Commands | /usr/bin/time | |
| tnchkdb | System Security | /usr/sbin/tnchkdb | privs |
| tnctl | System Security | /usr/sbin/tnctl | privs |
| tnd | boot | /usr/sbin/tnd | privs |
| tnd | System Security | /usr/sbin/tnd | privs |
| tninfo | System Security | /usr/sbin/tninfo | privs |

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| tokmapctl | Object Label Management | /usr/sbin/tokmapctl | privs |
| tokmapd | Maintenance and Repair | /usr/sbin/tokmapd | privs |
| touch | Basic Commands | /usr/bin/touch | |
| troff | Basic Commands | /usr/bin/troff | |
| true | Basic Commands | /usr/bin/true | |
| truss | Object Privilege Management | /usr/bin/truss | |
| tsolxagent | Basic Actions | /usr/dt/bin/tsolxagent | |
| ttsession | Basic Actions | /usr/dt/bin/ttsession | |
| tty | Basic Commands | /usr/bin/tty | |
| tunefs | Audit Control | /usr/sbin/tunefs | UID = 0, GID = 3, privs |
| tunefs | Maintenance and Repair | /usr/sbin/tunefs | UID = 0, privs |
| ufsdump | Media Backup | /usr/sbin/ufsdump | GID = 3, privs |
| ufsrestore | Media Restore | /usr/sbin/ufsrestore | privs |
| umount | Audit Control | /usr/sbin/umount | UID = 0, privs |
| umount | System Management | /usr/sbin/umount | UID = 0, privs |
| umountall | Audit Control | /usr/sbin/umountall | UID = 0, privs |
| umountall | System Management | /usr/sbin/umountall | privs |
| uname | Basic Commands | /usr/bin/uname | |
| uncompress | Basic Commands | /usr/bin/uncompress | |
| uniq | Basic Commands | /usr/bin/uniq | |
| unshare | Audit Control | /usr/sbin/unshare | UID = 0, privs |
| unshare | System Management | /usr/sbin/unshare | UID = 0, privs |
| unshareall | Audit Control | /usr/sbin/unshareall | privs |
| unshareall | System Management | /usr/sbin/unshareall | UID = 0, privs |
| usermgr | User Management | /opt/SUNWadm/2.1/bin/usermgr | privs |
| usermgr | User Security | /opt/SUNWadm/2.1/bin/usermgr | privs |
| vmstat | System Management | /usr/bin/vmstat | |

*Table A-2* Commands and Their Associated Execution Profiles

| Command | Profile | Path | Security Attributes |
|---|---|---|---|
| whereis | Basic Commands | /usr/ucb/whereis | |
| which | Basic Commands | /usr/bin/which | |
| who | Basic Commands | /usr/bin/who | |
| whoami | Basic Commands | /usr/ucb/whoami | |
| writeaudit | Audit Control | /usr/bin/writeaudit | privs |
| xhost | System Management | /usr/openwin/bin/xhost | privs |

## *Finding Actions in Execution Profiles*

Table A-3 lists each action contained in any execution profile and the execution profile(s) to which it is assigned. Remember that an action may be contained in more than one execution profile. The table also indicates any security attributes assigned to the action: minimum label, maximum label, setUID value, setGID value, and privileges. The term *privs* indicates that the action has one or more privileges. For specific privilege information, see the individual profile tables in Appendix B, "Profile Definition Tables," or use the Profile Manager to determine which privileges are actually applied to the action within that profile.

*Table A-3* Actions and Their Associated Execution Profiles

| Actions | Profiles | Security Attributes |
|---|---|---|
| AuditClass | Audit Control | min label = ADMIN_LOW, max label = ADMIN_LOW, privs |
| AuditControl | Audit Control | min label = ADMIN_LOW, max label = ADMIN_LOW, privs |
| AuditEvent | Audit Control | min label = ADMIN_LOW, max label = ADMIN_LOW, privs |
| AuditStartup | Audit Control | min label = ADMIN_LOW, max label = ADMIN_LOW, privs |

*Table A-3* Actions and Their Associated Execution Profiles

| Actions | Profiles | Security Attributes |
|---|---|---|
| AuditUser | Audit Control | min label = ADMIN_LOW, max label = ADMIN_LOW, privs |
| CheckEncodings | Object Label Management | privs |
| Dbmgr | System Management | privs |
| Dbmgr | System Security | privs |
| DNS_Resolve | System Management | privs |
| Dtappmgr | Basic Actions | |
| Dtcalc | Basic Actions | |
| Dtcm | Basic Actions | |
| Dtfile | Basic Actions | |
| Dtfile | Object Access Management | |
| Dtfile | Object Label Management | |
| Dtfile | Object Privilege Management | |
| DtfileHome | Basic Actions | |
| DtfileHome | Object Access Management | |
| DtfileHome | Object Label Management | |
| DtfileHome | Object Privilege Management | |
| Dthelpview | Basic Actions | |
| Dtmail | Basic Actions | |
| Dtmanpageview | Basic Actions | |
| Dtpad | Basic Actions | |
| Dtterm | Basic Actions | |
| Dttrash | Basic Actions | |
| Dttrash | Object Access Management | |

*Table A-3* Actions and Their Associated Execution Profiles

| Actions | Profiles | Security Attributes |
|---|---|---|
| Dttrash | Object Label Management | |
| DtTTMediaOpen | Basic Actions | |
| DtUnlink | Basic Actions | |
| EditEncodings | Object Label Management | privs |
| EditMotd | System Management | max label = ADMIN_LOW, privs |
| Groupmgr | User Management | |
| Hostmgr | System Management | |
| InvokeFILEMGR | Basic Actions | |
| InvokeFILEMGR | Object Access Management | |
| InvokeFILEMGR | Object Privilege Management | |
| InvokeMAILER | Basic Actions | |
| Nsswitch | System Security | privs |
| Open | Basic Actions | |
| OpenFolder | Basic Actions | |
| OWtapetool | Media Backup | |
| OWtapetool | Media Restore | |
| Print | Basic Actions | |
| Printermgr | System Security | |
| SDTimage | Basic Actions | |
| SendMail | System Security | privs |
| Serialmgr | System Security | |
| ShareFS | System Management | privs |
| Tar | Media Backup | |
| TarList | Media Backup | |

*Table A-3* Actions and Their Associated Execution Profiles

| Actions | Profiles | Security Attributes |
|---|---|---|
| TarList | Media Restore | |
| TarUnpack | Media Restore | |
| TextEditor | Basic Actions | |
| Trash | Basic Actions | |
| TrustedEditor | System Security | privs |
| Usermgr | User Management | privs |
| Vfstab | System Management | privs |
| Vfstab_adjunct | System Security | privs |
| Usermgr | User Management | privs |

# A

# Profile Definition Tables

B

This appendix contains tables that show:

- The names and purposes of all the execution profiles in the default system
- All the actions, authorizations, and commands defined for each execution profile shipped with the default system

This chapter contains the following sections.


| | |
|---|---|
| *Profile Names, Purposes, and Roles Tables* | *page 512* |
| *Profile Name: All* | *page 514* |
| *Profile Name: All Authorizations* | *page 515* |
| *Profile Name: Audit Control* | *page 516* |
| *Profile Name: Audit Review* | *page 519* |
| *Profile Name: Basic Actions* | *page 520* |
| *Profile Name: Basic Commands* | *page 522* |
| *Profile Name: Convenient Authorizations* | *page 525* |
| *Profile Name: Enable Login* | *page 526* |
| *Profile Name: Maintenance and Repair* | *page 527* |
| *Profile Name: Media Backup* | *page 528* |
| *Profile Name: Media Restore* | *page 530* |
| *Profile Name: NIS+ Administration* | *page 532* |
| *Profile Name: NIS+ Security Administration* | *page 533* |
| *Profile Name: Object Access Management* | *page 535* |

# B


| | |
|---|---|
| *Profile Name: Object Label Management* | *page 537* |
| *Profile Name: Object Privilege Management* | *page 539* |
| *Profile Name: Outside Accred* | *page 541* |
| *Profile Name: Privileged Shells* | *page 541* |
| *Profile Name: System Management* | *page 542* |
| *Profile Name: System Security* | *page 549* |
| *Profile Name: User Management* | *page 552* |
| *Profile Name: User Security* | *page 553* |
| *Profile Name: inetd* | *page 554* |
| *Profile Name: boot* | *page 557* |


## *Profile Names, Purposes, and Roles Tables*

*Table B-1* Name and Purpose of Profiles *(1 of 2)*

| Profile Name | Purpose |
|---|---|
| All | Give account access to all existing and potential commands and actions without security attributes, like a normal UNIX user |
| All Authorizations | Give account all authorizations |
| Audit Control | Administer the audit subsystem |
| Audit Review | View the audit trail |
| Basic Actions | Use common actions you need in CDE |
| Basic Commands | Use rudimentary commands no role can do without |
| Convenient Authorizations | Authorize account to enable logins, print a PostScript file, print without labels, and login remotely |
| Enable Login | Allow yourself and other users to log in after boot |
| Maintenance and Repair | Use commands needed to maintain or repair a system |
| Media Backup | Backup files |
| Media Restore | Restore files from backup |
| NIS+ Administration | Use NIS+ scripts/commands that are not security-related |
| NIS+ Security Administration | Use NIS+ security-related scripts/commands |
| Object Access Management | Change ownership and permissions on files |

*Table B-1* Name and Purpose of Profiles *(2 of 2)*

| Profile Name | Purpose |
|---|---|
| Object Label Management | Change labels of files and set up system-wide labels |
| Object Privilege Management | Install new applications, analyze use of privilege, and change privileges on executable files |
| Outside Accred | Operate outside system accreditation range |
| Privileged Shells | Run Borne, Korn, and C shell with all privileges. |
| System Management | Do general administrative tasks such as mounting file systems, editing system databases, managing print queues, and so forth |
| User Management | Manage aspects of creating or managing user attributes that are not security-related (complement of the “User Security” profile) |
| User Security | Create or modify a user’s security attributes (complement to the “User Manager” profile) |
| System Security | Do essential system security tasks such as setting labels on mounted file systems, trusted network configuration, and so forth |
| inetd | Privileges for programs executed by the inetd. Do not assign this profile to users. |
| boot | System start-up/shutdown profile. |

*Table B-2* Profiles Assigned to the Default Roles

| Security Administrator | System Administrator | Root | Operator |
|---|---|---|---|
| Audit Control | Audit Review | All | Basic Actions |
| Basic Actions | Basic Actions | All Authorizations | Basic Commands |
| Basic Commands | Basic Commands | Privileged Shells | Outside Accred |
| NIS+ Security Administration | Enable Login | NIS+ Security Administration | Media Backup |
| Object Access Management | | User Security | |
| Object Label Management | | | |
| Object Privilege Management | | | |
| Outside Accred | | | |
| System Security | | | |
| User Security | | | |

# B

*Table B-3* Common Profiles (in More Than One Role

| Security Administrator | System Administrator | System Operator |
|---|---|---|
| Basic Actions | Basic Actions | Basic Actions |
| Basic Commands | Basic Commands | Basic Commands |
| Outside Accred | Outside Accred | Outside Accred |

*Table B-4* Profiles That Divide Responsibilities Among Roles

| Security Administrator | System Administrator | System Operator | Root |
|---|---|---|---|
| Audit Control | Audit Review | | |
| NIS+ Security Administration | NIS+ Administration | | NIS+ Security Administration |
| System Security | System Management | | |
| User Security | User Management | | User Security |
| | Media Restore | Media Backup | |

*Table B-5* Profiles Unique to One Role

| Security Administrator | System Administrator | Root |
|---|---|---|
| Object Access Management | Enable Login | All |
| Object Label Management | Maintenance and Repair | All Authorizations |
| Object Privilege Management | Media Restore | Privileged Shells |

## *Profile Name: All*

Definition: all commands and all actions (to set up a normal UNIX user). No authorizations. All checks for commands and actions are disabled, so any command or action, existing or future) can be used without special attributes, such as privileges.

*Table B-6* Actions Defined for the All Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| *** (All existing and potential)** | none | none | none | none | none |

*Table B-7* Authorizations Defined for the All Profile

| Authorizations |
|---|
| none |

*Table B-8* Commands Defined for the All Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| ***/ (All existing and potential)** | none | none | none | none | none |

## *Profile Name: All Authorizations*

Definition: All authorizations in the system

*Table B-9* Actions Defined for the All Authorizations Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-10* Authorizations Defined for the All Authorizations Profile

| Authorizations |
|---|
| all |

*Table B-11* Commands Defined for the All Authorizations Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

## *Profile Name: Audit Control*

Definition: Administer the audit subsystem.

*Table B-12* Actions Defined for the Audit Control Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| AuditClass | ADMIN_LOW | ADMIN_LOW | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |
| AuditControl | ADMIN_LOW | ADMIN_LOW | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |
| AuditEvent | ADMIN_LOW | ADMIN_LOW | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |
| AuditStartup | ADMIN_LOW | ADMIN_LOW | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |
| AuditUser | ADMIN_LOW | ADMIN_LOW | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |

*Table B-13* Authorizations Defined for the Audit Control Profile

| Authorizations |
|---|
| set user audit flags |
| set/get file audit flags |

*Table B-14* Commands Defined for the Audit Control Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /etc/security/bsmconv | ADMIN_LOW | ADMIN_LOW | 0 | none | none |
| /etc/security/bsmunconv | ADMIN_LOW | ADMIN_LOW | 0 | none | none |
| /usr/bin/mkdir | none | none | none | none | file_dac_write |
| /usr/bin/writeaudit | none | none | none | none | proc_audit_appl<br>proc_audit_tcb |
| /usr/sbin/audit | none | none | 0 | none | file_mac_read<br>proc_audit_tcb<br>oroc_mac_write<br>sys_audit |
| /usr/sbin/auditconfig | ADMIN_LOW | ADMIN_LOW | 0 | none | sys_audit |
| /usr/sbin/auditd | none | none | 0 | none | file_mac_write<br>proc_set_clr<br>oroc_set_il<br>oroc_set_sl<br>sys_audit |
| /usr/sbin/auditstat | none | none | 0 | none | none |
| /usr/sbin/format | none | none | 0 | none | all |
| /usr/sbin/mkfs | none | none | none | none | all |

*Table B-14* Commands Defined for the Audit Control Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/mount | none | none | 0 | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>net_privaddr<br>proc_setil<br>proc_setsl<br>sys_mount<br>sys_trans_label |
| /usr/sbin/mountall | none | none | 0 | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>net_privaddr<br>proc_setil<br>proc_setsl<br>sys_mount<br>sys_trans_label |
| /usr/sbin/newfs | none | none | 0 | none | all |
| /usr/sbin/newsecfs | none | none | none | none | all |
| /usr/sbin/share | none | none | 0 | none | sys_nfs |
| /usr/sbin/shareall | none | none | 0 | none | sys_nfs |
| /usr/sbin/tunefs | none | none | 0 | 3 | none |

*Table B-14* Commands Defined for the Audit Control Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/umount | none | none | 0 | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>net_privaddr<br>proc_setil<br>proc_setsl<br>sys_mount<br>sys_trans_label |
| /usr/sbin/umountall | none | none | 0 | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>net_privaddr<br>proc_setil<br>proc_setsl<br>sys_mount<br>sys_trans_label |
| /usr/sbin/unshare | none | none | 0 | none | sys_nfs |
| /usr/sbin/unshareall | none | none | 0 | none | sys_nfs |

## *Profile Name: Audit Review*

Definition: View the audit trail

*Table B-15* Actions Defined for the Audit Review Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-16* Authorizations Defined for the Audit Review Profile

| Authorizations |
|---|
| none |

*Table B-17* Commands Defined for the Audit Review Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/awk | ADMIN_HIGH | ADMIN_HIGH | 0 | none | none |
| /usr/bin/cat | ADMIN_HIGH | ADMIN_HIGH | 0 | none | none |
| /usr/bin/grep | ADMIN_HIGH | ADMIN_HIGH | 0 | none | none |
| /usr/bin/sed | ADMIN_HIGH | ADMIN_HIGH | 0 | none | none |
| /usr/bin/tail | ADMIN_HIGH | ADMIN_HIGH | 0 | none | none |
| /usr/sbin/auditreduce | ADMIN_HIGH | none | 0 | none | file_dac_read sys_audit |
| /usr/sbin/praudit | ADMIN_HIGH | none | 0 | none | file_dac_read sys_audit |

## *Profile Name: Basic Actions*

Definition: The common actions you need in CDE

*Table B-18* Actions Defined for the Basic Actions Profile *(1 of 2)*

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Dtcalc | none | none | none | none | none |
| Dtcm | none | none | none | none | none |
| Dtfile | none | none | none | none | none |
| Dthelpview | none | none | none | none | none |
| SDTimage | none | none | none | none | none |
| Dtmail | none | none | none | none | none |

*Table B-18* Actions Defined for the Basic Actions Profile *(2 of 2)*

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Dtmanpageview | none | none | none | none | none |
| Dtterm | none | none | none | none | none |
| Dtpad | none | none | none | none | none |
| Dttrash | none | none | none | none | none |
| Print | none | none | none | none | none |
| Dtappmgr | none | none | none | none | none |
| DtTTMediaOpen | none | none | none | none | none |
| DtfileHome | none | none | none | none | none |
| InvokeFILEMGR | none | none | none | none | none |
| InvokeMAILER | none | none | none | none | none |
| OpenFolder | none | none | none | none | none |
| DtUnlink | none | none | none | none | none |
| Open | none | none | none | none | none |
| TextEditor | none | none | none | none | none |
| Trash | none | none | none | none | none |

*Table B-19* Authorizations Defined for the Basic Actions Profile

| Authorizations |
|---|
| none |

*Table B-20* Commands Defined for the Basic Actions Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/dt/bin/tsolxagent | none | none | none | none | none |
| /usr/dt/bin/ttsession | none | none | none | none | none |

# B

## *Profile Name: Basic Commands*

Definition: Use rudimentary commands no role can do without

*Table B-21* Actions Defined for the Basic Commands Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-22* Authorizations Defined for the Basic Commands Profile

| Authorizations |
|---|
| none |

*Table B-23* Commands Defined for the Basic Commands Profile *(1 of 4)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/adminvi | none | none | none | none | none |
| /usr/bin/awk | none | none | none | none | none |
| /usr/bin/cat | none | none | none | none | none |
| /usr/bin/cd | none | none | none | none | none |
| /usr/bin/chmod | none | none | none | none | none |
| /usr/bin/clear | none | none | none | none | none |
| /usr/bin/cmp | none | none | none | none | none |
| /usr/bin/col | none | none | none | none | none |
| /usr/bin/compress | none | none | none | none | none |
| /usr/bin/cp | none | none | none | none | none |
| /usr/bin/cut | none | none | none | none | none |
| /usr/bin/df | none | none | none | none | none |
| /usr/bin/diff | none | none | none | none | none |
| /usr/bin/diff3 | none | none | none | none | none |

*Table B-23* Commands Defined for the Basic Commands Profile *(2 of 4)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/dircmp | none | none | none | none | none |
| /usr/bin/dirname | none | none | none | none | none |
| /usr/bin/du | none | none | none | none | none |
| /usr/bin/echo | none | none | none | none | none |
| /usr/bin/env | none | none | none | none | none |
| /usr/bin/expr | none | none | none | none | none |
| /usr/bin/false | none | none | none | none | none |
| /usr/bin/fgrep | none | none | none | none | none |
| /usr/bin/file | none | none | none | none | none |
| /usr/bin/fold | none | none | none | none | none |
| /usr/bin/getlabel | none | none | none | none | none |
| /usr/bin/grep | none | none | none | none | none |
| /usr/bin/head | none | none | none | none | none |
| /usr/bin/hostid | none | none | none | none | none |
| /usr/bin/hostname | none | none | none | none | none |
| /usr/bin/id | none | none | none | none | none |
| /usr/bin/join | none | none | none | none | none |
| /usr/bin/ldd | none | none | none | none | none |
| /usr/bin/ln | none | none | none | none | none |
| /usr/bin/look | none | none | none | none | none |
| /usr/bin/lp | none | none | none | none | none |
| /usr/bin/ls | none | none | none | none | none |
| /usr/bin/mailq | none | none | none | 2 | none |
| /usr/bin/man | none | none | none | none | none |
| /usr/bin/mkdir | none | none | none | none | none |
| /usr/bin/more | none | none | none | none | none |
| /usr/bin/mv | none | none | none | none | none |

*Table B-23* Commands Defined for the Basic Commands Profile *(3 of 4)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/niscat | none | none | none | none | none |
| /usr/bin/nisdefaults | none | none | none | none | none |
| /usr/bin/niserror | none | none | none | none | none |
| /usr/bin/nisgrep | none | none | none | none | none |
| /usr/bin/nismatch | none | none | none | none | none |
| /usr/bin/nistest | none | none | none | none | none |
| /usr/bin/nroff | none | none | none | none | none |
| /usr/bin/page | none | none | none | none | none |
| /usr/bin/pfsh | none | none | none | none | none |
| /usr/bin/pg | none | none | none | none | none |
| /usr/bin/pr | none | none | none | none | none |
| /usr/bin/pwd | none | none | none | none | none |
| /usr/bin/rcp | none | none | none | none | none |
| /usr/bin/rlogin | none | none | none | none | none |
| /usr/bin/rm | none | none | none | none | none |
| /usr/bin/rmdir | none | none | none | none | none |
| /usr/bin/script | none | none | none | none | none |
| /usr/bin/sdiff | none | none | none | none | none |
| /usr/bin/sleep | none | none | none | none | none |
| /usr/bin/sort | none | none | none | none | none |
| /usr/bin/spell | none | none | none | none | none |
| /usr/bin/stty | none | none | none | none | none |
| /usr/bin/tail | none | none | none | none | none |
| /usr/bin/tbl | none | none | none | none | none |
| /usr/bin/test | none | none | none | none | none |
| /usr/bin/tfind | none | none | none | none | none |
| /usr/bin/time | none | none | none | none | none |

*Table B-23* Commands Defined for the Basic Commands Profile *(4 of 4)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/touch | none | none | none | none | none |
| /usr/bin/troff | none | none | none | none | none |
| /usr/bin/true | none | none | none | none | none |
| /usr/bin/tty | none | none | none | none | none |
| /usr/bin/uname | none | none | none | none | none |
| /usr/bin/uncompress | none | none | none | none | none |
| /usr/bin/uniq | none | none | none | none | none |
| /usr/bin/which | none | none | none | none | none |
| /usr/bin/who | none | none | none | none | none |
| /usr/ucb/rsh | none | none | none | none | none |
| /usr/ucb/whereis | none | none | none | none | none |
| /usr/ucb/whoami | none | none | none | none | none |

## *Profile Name: Convenient Authorizations*

Definition: Authorize account to enable logins, print a PostScript file, print without labels, and login remotely.

*Table B-24* Actions Defined for the Convenient Authorizations Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| * | none | none | none | none | none |

*Table B-25* Authorizations Defined for the Convenient Authorizations Profile

| Authorizations |
|---|
| enable logins |
| tsol_auth_reserved13 |
| print a PostScript file |
| print without labels |
| remote login |
| set application search path |

*Table B-26* Commands Defined for the Convenient Authorizations Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/ucb | none | none | none | none | none |

## *Profile Name: Enable Login*

Definition: Allow yourself and other users to log in after boot

*Table B-27* Actions Defined for the Enable Login Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-28* Authorizations Defined for the Enable Login Profile

| Authorizations |
|---|
| enable logins |

*Table B-29* Commands Defined for the Enable Login Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/ucb | none | none | none | none | none |

## *Profile Name: Maintenance and Repair*

Definition: Use commands needed to maintain or repair a system

*Table B-30* Actions Defined for the Maintenance and Repair Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-31* Authorizations Defined for the Maintenance and Repair Profile

| Authorizations |
|---|
| enable logins<br>remote login<br>terminal login |

*Table B-32* Commands Defined for the Maintenance and Repair Profile *(1 of 2)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /etc/autopush | none | none | none | none | none |
| /etc/clri | none | none | none | none | none |
| /usr/bin/adb | none | none | none | none | none |
| /usr/bin/netstat | none | none | none | none | none |
| /usr/sbin/add_drv | none | none | none | none | none |
| /usr/sbin/aspppd | none | none | none | none | none |
| /usr/sbin/cachefslog | none | none | none | none | none |

*Table B-32* Commands Defined for the Maintenance and Repair Profile *(2 of 2)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/clri | none | none | none | none | none |
| /usr/sbin/crash | none | none | none | none | none |
| /usr/sbin/eeprom | none | none | none | none | none |
| /usr/sbin/fsck | none | none | none | none | none |
| /usr/sbin/fsdb | none | none | none | none | none |
| /usr/sbin/fsirand | none | none | none | none | none |
| /usr/sbin/grpck | none | none | none | none | none |
| /usr/sbin/halt | none | none | 0 | none | none |
| /usr/sbin/ncheck | none | none | none | none | none |
| /usr/sbin/ping | none | none | none | none | none |
| /usr/sbin/poweroff | none | none | 0 | none | sys_boot |
| /usr/sbin/prtconf | none | none | none | none | none |
| /usr/sbin/pwck | none | none | none | none | none |
| /usr/sbin/reboot | none | none | 0 | none | sys_boot |
| /usr/sbin/rem_drv | none | none | none | none | none |
| /usr/sbin/snoop | none | none | none | none | none |
| /usr/sbin/spray | none | none | none | none | none |
| /usr/sbin/strace | none | none | none | none | none |
| /usr/sbin/syslogd | none | none | none | none | none |
| /usr/sbin/tokmapd | none | none | none | none | net_privaddr<br>proc_setclr<br>proc_setsl |
| /usr/sbin/tunefs | none | none | 0 | none | all |

## *Profile Name: Media Backup*

Definition: Back up files

*Table B-33* Actions Defined for the Media Backup Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Tar | none | none | none | none | none |
| TarList | none | none | none | none | none |
| OWtapetool | none | none | none | none | none |

*Table B-34* Authorizations Defined for the Media Backup Profile

| Authorizations |
|---|
| allocate device |

*Table B-35* Commands Defined for the Media Backup Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/mt | none | none | none | none | none |
| /usr/bin/tar | none | none | none | none | file_audit<br>file_dac_read<br>file_dac_search<br>file_downgrade_il<br>file_downgrade_sl<br>file_mac_read<br>file_mac_write<br>file_upgrade_il<br>file_upgrade_sl<br>sys_trans_label |
| /usr/sbin/ufsdump | none | none | none | 3 | file_audit<br>file_dac_read<br>file_dac_search<br>file_downgrade_il<br>file_downgrade_sl<br>file_mac_read<br>file_mac_write<br>sys_trans_label |

# B

## *Profile Name: Media Restore*

Definition: Restore files from backup

*Table B-36* Actions Defined for the Media Restore Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| TarList | none | none | none | none | none |
| TarUnpack | none | none | none | none | none |
| OWtapetool | none | none | none | none | none |

*Table B-37* Authorizations Defined for the Media Restore Profile

| Authorizations |
|---|
| allocate device |

*Table B-38* Commands Defined for the Media Restore Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/cpio | none | none | none | none | none |
| /usr/bin/mt | none | none | none | none | none |
| /usr/bin/tar | none | none | none | none | file_audit<br>file_chown<br>file_dac_read<br>file_dac_search<br>file_dac_write<br>file_downgrade_il<br>file_downgrade_sl<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>file_owner<br>file_setdac<br>file_setid<br>file_setpriv<br>file_upgrade_il<br>file_upgrade_sl<br>sys_devices<br>sys_trans_label |
| /usr/sbin/ufsrestore | none | none | none | none | file_audit<br>file_chown<br>file_dac_read<br>file_dac_search<br>file_dac_write<br>file_downgrade_il<br>file_downgrade_sl<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>file_nofloat<br>file_owner<br>file_setdac<br>file_setid<br>file_setpriv<br>file_upgrade_il<br>file_upgrade_sl<br>sys_trans_label |

# B

## *Profile Name: NIS+ Administration*

Definition: Use NIS+ scripts/commands that are not security-related

*Table B-39* Actions Defined for the NIS+ Administration Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-40* Authorizations Defined for the NIS+ Administration Profile

| Authorizations |
|---|
| none |

*Table B-41* Commands Defined for the NIS+ Administration Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/nischttl | none | none | none | none | none |
| /usr/bin/nisln | none | none | none | none | none |
| /usr/lib/nis/nisctl | none | none | none | none | none |
| /usr/lib/nis/nisping | none | none | none | none | none |
| /usr/lib/nis/nisshowcache | none | none | none | none | none |
| /usr/lib/nis/nisstat | none | none | none | none | none |
| /usr/lib/nis/nistnsetup | none | none | none | none | none |
| /usr/lib/nis/nistntime | none | none | none | none | none |
| /usr/sbin/nscd | none | none | none | none | file_dac_write<br>file_setid<br>net_mac_read<br>net_upgrade_sl<br>proc_dumpcore<br>proc_nofloat<br>proc_setclr<br>sys_net_config<br>sys_trans_label |

## *Profile Name: NIS+ Security Administration*

Definition: Use NIS+ security-related scripts/commands

*Table B-42* Actions Defined for the NIS+ Security Administration Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-43* Authorizations Defined for the NIS+ Security Administration Profile

| Authorizations |
|---|
| none |

*Table B-44* Commands Defined for the NIS+ Security Administration Profile *(1 of 2)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/chkey | none | none | none | none | none |
| /usr/bin/nisaddcred | none | none | none | none | none |
| /usr/bin/nischgrp | none | none | none | none | none |
| /usr/bin/nischmod | none | none | none | none | none |
| /usr/bin/nischown | none | none | none | none | none |
| /usr/bin/nisgrpadm | none | none | none | none | none |
| /usr/bin/nismkdir | none | none | none | none | none |
| /usr/bin/nispasswd | none | none | none | none | none |
| /usr/bin/nisrm | none | none | none | none | none |
| /usr/bin/nisrmdir | none | none | none | none | none |
| /usr/bin/nistbladm | none | none | none | none | none |
| /usr/lib/nisaddent | none | none | none | none | none |

*Table B-44* Commands Defined for the NIS+ Security Administration Profile *(2 of 2)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/lib/nisclient | none | none | none | none | file_dac_read<br>file_dac_write<br>file_mac_read<br>net_mac_read<br>net_reply_equal<br>net_setclr<br>net_setid<br>net_setpriv<br>proc_nofloat<br>sys_net_config |
| /usr/lib/nispopulate | none | none | none | none | file_dac_read<br>file_mac_read<br>net_mac_read<br>net_reply_equal<br>net_setclr<br>net_setid<br>net_setpriv<br>net_upgrade_sl<br>proc_nofloat<br>proc_setclr<br>proc_setsl<br>sys_net_config |
| /usr/lib/nisserver | none | none | none | none | none |
| /usr/lib/nis/nissetup | none | none | none | none | none |
| /usr/lib/nis/nisupdkeys | none | none | none | none | none |
| /usr/sbin/newkey | none | none | none | none | none |
| /usr/sbin/nisinit | none | none | none | none | none |
| /usr/sbin/nislog | none | none | none | none | none |

## *Profile Name: Nothing*

Definition: Do not allow the account to do anything except log in

*Table B-45* Actions Defined for the Enable Login Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-46* Authorizations Defined for the Enable Login Profile

| Authorizations |
|---|
| none |

*Table B-47* Commands Defined for the Enable Login Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

## *Profile Name: Object Access Management*

Definition: Change ownership and permissions on files

*Table B-48* Actions Defined for the Object Access Management Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Dtfile | none | none | none | none | none |
| Dttrash | none | none | none | none | none |
| DtfileHome | none | none | none | none | none |
| InvokeFILEMGR | none | none | none | none | none |

*Table B-49* Authorizations Defined for the Object Access Management Profile

| Authorizations |
|---|
| act as file owner |
| change file owner |

*Table B-50* Commands Defined for the Object Access Management Profile *(1 of 2)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/chgrp | none | none | none | none | file_chown<br>file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_owner<br>file_setdac<br>file_setid |
| /usr/bin/chmod | none | none | none | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_setdac<br>file_setid |
| /usr/bin/chown | none | none | none | none | file_chown<br>file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_owner |
| /usr/bin/getfacl | none | none | none | none | file_dac_search<br>file_mac_read<br>file_mac_search |
| /usr/bin/getfattrflag | none | none | none | none | file_audit<br>file_dac_search<br>file_mac_read<br>file_mac_search |

*Table B-50* Commands Defined for the Object Access Management Profile *(2 of 2)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/getlabel | none | none | none | none | file_dac_search<br>file_mac_read<br>file_mac_search |
| /usr/bin/setfacl | none | none | none | none | file_dac_read<br>file_dac_search<br>file_dac_write<br>file_mac_read<br>file_mac_search<br>file_setdac |
| /usr/bin/setfattrflag | none | none | none | none | file_audit<br>file_dac_search<br>file_mac_search<br>file_mac_write<br>file_owner |

## *Profile Name: Object Label Management*

Definition: Change labels of files and set up system-wide labels.

*Table B-51* Actions Defined for the Object Label Management Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Dtfile | none | none | none | none | none |
| Dttrash | none | none | none | none | none |
| CheckEncodings | none | none | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb<br>sys_trans_label |
| EditEncodings | none | none | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |
| DtfileHome | none | none | none | none | none |

*Table B-52* Authorizations Defined for the Object Label Management Profile

| Authorizations |
|---|
| allocate device |
| bypass view of file contents on drag and drop |
| downgrade file sensitivity label |
| upgrade file sensitivity label |
| use all defined labels |

*Table B-53* Commands Defined for the Object Label Management Profile *(1 of 2)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/cp | none | none | none | none | file_mac_write |
| /usr/bin/getlabel | none | none | none | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>sys_trans_label |
| /usr/bin/getmldadorn | none | none | none | none | none |
| /usr/bin/getsldname | none | none | none | none | none |
| /usr/bin/mldpwd | none | none | none | none | none |
| /usr/bin/mldrealpath | none | none | none | none | none |
| /usr/bin/mv | none | none | none | none | file_mac_write |
| /usr/bin/rm | none | none | none | none | file_dac_read<br>file_dac_search<br>file_dac_write<br>file_mac_write |
| /usr/bin/setfattrflag | none | none | none | none | none |

*Table B-53* Commands Defined for the Object Label Management Profile *(2 of 2)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/setlabel | none | none | none | none | file_dac_read<br>file_dac_search<br>file_dac_write<br>file_downgrade_il<br>file_downgrade_sl<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>file_nofloat<br>file_owner<br>file_upgrade_il<br>file_upgrade_sl |
| /usr/bin/tfind | none | none | none | none | file_dac_read<br>file_dac_search |
| /usr/proc/bin/plabel | none | none | none | none | proc_mac_read<br>sys_trans_label |
| /usr/sbin/atohexlabel | none | none | none | none | none |
| /usr/sbin/chk_encodings | none | none | none | none | none |
| /usr/sbin/hextoalabel | none | none | none | none | sys_trans_label |
| /usr/sbin/tokmapctl | none | none | none | none | net_mac_read<br>net_privaddr |

## *Profile Name: Object Privilege Management*

Definition: Install new applications, analyze user of privilege, and change privileges on executable files

*Table B-54* Actions Defined for the Object Privilege Management Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Dtfile | none | none | none | none | none |
| DtfileHome | none | none | none | none | none |
| InvokeFILEMGR | none | none | none | none | none |

*Table B-55* Authorizations Defined for the Object Privilege Management Profile

| Authorizations |
|---|
| set file privileges |

*Table B-56* Commands Defined for the Object Privilege Management Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/adb | none | none | none | none | none |
| /usr/bin/getfpriv | none | none | none | none | none |
| /usr/bin/kill | none | none | none | none | none |
| /usr/bin/ldd | none | none | none | none | none |
| /usr/bin/pkginfo | none | none | none | none | none |
| /usr/bin/setfpriv | none | none | none | none | file_dac_read<br>file_dac_search<br>file_dac_write<br>file_mac_read<br>file_mac_search<br>file_owner<br>file_setid<br>file_setpriv |
| /usr/bin/testfpriv | none | none | none | none | file_dac_search<br>file_mac_read<br>file_mac_search |
| /usr/bin/truss | none | none | none | none | none |
| /usr/proc/bin/ppriv | none | none | none | none | proc_mac_read<br>proc_owner |
| /usr/proc/bin/pprivtest | none | none | none | none | proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/sbin/pkgadd | none | none | none | none | none |
| /usr/sbin/pkgchk | none | none | none | none | none |

*Table B-56* Commands Defined for the Object Privilege Management Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/pkgrm | none | none | none | none | none |
| /usr/sbin/runpd | none | none | none | none | none |
| /usr/sbin/swmtool | none | none | none | none | none |

## *Profile Name: Outside Accred*

Definition: Allow user to operate outside user accreditation range

*Table B-57* Actions Defined for the Outside Accred Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-58* Authorizations Defined for the Outside Accred Profile

| Authorizations |
|---|
| use all defined labels |

*Table B-59* Commands Defined for the Outside Accred Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/ | none | none | none | none | none |

## *Profile Name: Privileged Shells*

Definition: shells that run with all privilege

*Table B-60* Actions Defined for the Privileged Shells Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-61* Authorizations Defined for the Privileged Shells Profile

| Authorizations |
|---|
| none |

*Table B-62* Commands Defined for the Privileged Shells Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/csh | none | none | none | none | all |
| /usr/bin/ksh | none | none | none | none | all |
| /usr/bin/sh | none | none | none | none | all |

## *Profile Name: System Management*

Definition: Do general administrative tasks such as mounting file systems, editing system databases, managing print queues, and so forth

*Table B-63* Actions Defined for the System Management Profile *(1 of 2)*

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Dbmgr | none | none | none | none | sys_trans_label |
| Hostmgr | none | none | none | none | none |
| DNS_Resolve | none | none | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |

*Table B-63* Actions Defined for the System Management Profile *(2 of 2)*

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| EditMotd | none | ADMIN_LOW | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |
| Vfstab | none | none | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |
| ShareFS | none | none | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |

*Table B-64* Authorizations Defined for the System Management Profile

| Authorizations |
|---|
| administer printing |
| modify at users |
| modify bootparams |
| modify cron users |
| modify ethers |
| modify hosts |
| modify locale |
| modify netmasks |
| modify networks |
| modify protocols |
| modify rpc |
| modify services |
| modify timezone |

# B

*Table B-65* Commands Defined for the System Management Profile *(1 of 6)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/adminvi | none | none | none | none | none |
| /usr/bin/cancel | none | none | 71 | none | file_mac_read<br>file_mac_write |
| /usr/bin/date | none | none | none | none | sys_config |
| /usr/bin/disable | none | none | none | none | sys_devices |
| /usr/bin/enable | none | none | none | none | sys_devices |
| /usr/bin/kill | none | none | none | none | proc_mac_write<br>proc_owner |
| /usr/bin/lp | none | none | none | none | none |
| /usr/bin/lpstat | none | ADMIN_LOW | none | none | file_dac_read<br>file_mac_read |
| /sur/bin/mailq | none | none | none | 2 | file_dac_read<br>file_mac_read |
| /usr/bin/newaliases | none | none | 0 | none | net_mac_read<br>net_privaddr<br>proc_nofloat<br>proc_setil |
| /usr/bin/nice | none | none | none | none | none |
| /usr/bin/ps | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/bin/rdist | none | none | none | none | none |
| /usr/bin/renice | none | none | none | none | proc_mac_write<br>proc_owner |
| /usr/bin/rup | none | none | none | none | none |
| /usr/bin/vmstat | none | none | none | none | none |
| /usr/openwin/bin/xhost | none | none | none | none | win_config |

*Table B-65* Commands Defined for the System Management Profile *(2 of 6)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/proc/bin/pattr | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_owner |
| /usr/proc/bin/pclear | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_owner<br>sys_trans_label |
| /usr/proc/bin/pcred | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/pfiles | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/pflags | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/plabel | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner<br>sys_trans_label |
| /usr/proc/bin/pldd | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/pmap | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/prun | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |

*Table B-65* Commands Defined for the System Management Profile *(3 of 6)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/proc/bin/psig | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/pstack | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/pstop | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/ptime | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/ptree | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/pwait | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/proc/bin/pwdx | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/sbin/accept | none | none | none | none | sys_devices |
| /usr/sbin/allocate | none | none | none | none | file_chown<br>file_set_dac |
| /usr/sbin/deallocate | none | none | none | none | file_chown<br>file_set_dac |
| /usr/snin/format | none | none | 0 | none | all |

*Table B-65* Commands Defined for the System Management Profile *(4 of 6)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/getfsattr | none | none | none | none | file_dac_read<br>file_dac_search<br>file_mac_search |
| /usr/sbin/in.named | ADMIN_LOW | ADMIN_HIGH | 0 | none | file_mac_read<br>net_mac_read<br>net_privaddr<br>net_upgrade_sl<br>proc_dumpcore<br>proc_nofloat<br>proc_setclr<br>sys_config<br>sys_net_config<br>sys_trans_label |
| /usr/sbin/init | none | none | none | none | all |
| /usr/sbin/list_devices | none | none | none | none | none |
| /usr/sbin/lpadmin | none | ADMIN_LOW | 0 | none | none |
| /usr/sbin/lpfilter | none | ADMIN_LOW | 0 | none | none |
| /usr/sbin/lpmove | none | ADMIN_LOW | 0 | none | none |
| /usr/sbin/lpshut | none | none | 0 | none | none |
| /usr/sbin/lpsystem | none | ADMIN_LOW | 0 | none | none |
| /usr/sbin/lpusers | none | ADMIN_LOW | 0 | none | none |
| /usr/sbin/mkfile | none | none | 0 | none | none |
| /usr/sbin/mkfs | none | none | 0 | none | all |
| /usr/sbin/mount | none | none | 0 | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>net_privaddr<br>proc_setil<br>proc_setsl<br>sys_mount<br>sys_trans_label |

*Table B-65* Commands Defined for the System Management Profile *(5 of 6)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/mountall | none | none | 0 | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>net_priv_addr<br>proc_setil<br>proc_setsl<br>sys_mount<br>sys_trans_label |
| /usr/sbin/newfs | none | none | 0 | none | all |
| /usr/sbin/ping | none | none | none | none | none |
| /usr/sbin/reject | none | none | none | none | sys_devices |
| /usr/sbin/share | none | none | 0 | none | sys_nfs |
| /usr/sbin/shareall | none | none | 0 | none | sys_nfs |
| /usr/sbin/showmount | none | none | none | none | none |
| /usr/sbin/swap | none | none | none | none | sys_mount |
| /usr/sbin/umount | none | none | 0 | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>net_privaddr<br>proc_setil<br>proc_setsl<br>sys_mount<br>sys_trans_label |

*Table B-65* Commands Defined for the System Management Profile *(6 of 6)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/umountall | none | none | none | none | file_dac_read<br>file_dac_search<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>net_privaddr<br>proc_setil<br>proc_setsl<br>sys_mount<br>sys_trans_label |
| /usr/sbin/unshare | none | none | 0 | none | sys_nfs |
| /usr/sbin/unshareall | none | none | 0 | none | sys_nfs |

## *Profile Name: System Security*

Definition: Do essential system security tasks such as setting labels on mounted file systems, trusted network configuration, and so forth

*Table B-66* Actions Defined for the System Security Profile *(1 of 2)*

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Dbmgr | none | none | none | none | sys_trans_label |
| Printermgr | none | none | none | none | none |
| Serialmgr | none | none | none | none | none |
| TrustedEditor | none | none | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |

*Table B-66* Actions Defined for the System Security Profile *(2 of 2)*

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Nsswitch | none | none | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |
| SendMail | none | none | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |
| Vfstab_adjunct | none | none | none | none | file_dac_read<br>file_dac_write<br>proc_audit_appl<br>proc_audit_tcb |

*Table B-67* Authorizations Defined for the System Security Profile

| Authorizations |
|---|
| modify at admin |
| modify cron admin |
| modify netgroup |
| modify tnidb |
| modify tnrhdb |
| modify tnrhtp |
| print a PostScript file |
| print without labels |
| remote login |

*Table B-68* Commands Defined for the System Security Profile *(1 of 3)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/adminvi | none | none | none | none | none |
| /usr/bin/disable | none | none | none | none | sys_devices |
| /usr/bin/enable | none | none | none | none | sys_devices |

*Table B-68* Commands Defined for the System Security Profile *(2 of 3)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/bin/ps | none | none | none | none | file_dac_read<br>proc_mac_read<br>proc_nofloat<br>proc_owner |
| /usr/sbin/accept | none | none | none | none | sys_devices |
| /usr/sbin/add_drv | none | none | none | none | none |
| ‘/usr/sbin/allocate | none | none | none | none | file_chown<br>file_setdac |
| /usr/sbin/autopush | none | none | none | none | none |
| /usr/sbin/deallocate | none | none | none | none | file_chown<br>file_setdac |
| /usr/sbin/drvconfig | none | none | none | none | none |
| /usr/sbin/format | none | none | 0 | none | all |
| /usr/sbin/getfsattr | none | none | none | 3 | none |
| /usr/sbin/mkfs | none | none | 0 | none | all |
| /usr/bin/newfs | none | none | 0 | none | all |
| /usr/sbin/newsecfs | none | none | 0 | none | all |
| /usr/sbin/ping | none | none | none | none | none |
| /usr/sbin/reject | none | none | none | none | sys_devices |
| /usr/sbin/rem_drv | none | none | none | none | sys_devices |
| /usr/sbin/setfsattr | none | none | none | none | file_dac_write<br>file_downgrade_il<br>file_downgrade_sl<br>file_mac_search<br>file_setdac<br>file_setid<br>file_setpriv<br>file_upgrade_il<br>file_upgrade_sl |
| /usr/sbin/tnchkdb | none | none | none | none | file_mac_read<br>sys_trans_label |

*Table B-68* Commands Defined for the System Security Profile *(3 of 3)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/tnctl | none | none | none | none | sys_net_config<br>sys_trans_label |
| /usr/sbin/tnd | none | none | none | none | net_downgrade_sl<br>net_mac_read<br>net_privaddr<br>proc_setclr<br>proc_setsl<br>sys_net_config |
| /usr/sbin/tninfo | none | none | none | none | file_dac_read<br>file_mac_read<br>sys_net_config<br>sys_trans_label |

## *Profile Name: User Management*

Definition: Manage aspects of creating or managing user attributes that are not security-related. It is the complement of the “User Security” profile

*Table B-69* Actions Defined for the User Management Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Groupmgr | none | none | none | none | none |
| Usermgr | none | none | none | none | sys_trans_label |

*Table B-70* Authorizations Defined for the User Management Profile

| Authorizations |
|---|
| modify aliases<br>set attributes related to home directories<br>set user identity |

*Table B-71* Commands Defined for the User Management Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /opt/SUNWadm/2.1/bin/groupmgr | none | none | none | none | none |
| /opt/SUNWadm/2.1/bin/usermgr | none | none | none | none | sys_trans_label |

## *Profile Name: User Security*

Definition: Create or modify a user's security attributes. This profile is the complement to the "User Manager" profile

*Table B-72* Actions Defined for the User Security Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| Dbmgr | none | none | none | none | sys_trans_label |
| Profmgr | none | none | none | none | sys_trans_label,<br>win_dac_read<br>win_mac_read |
| Usermgr | none | none | none | none | sys_trans_label |

*Table B-73* Authorizations Defined for the User Security Profile

| Authorizations |
|---|
| modify auto_home |
| set idle time |
| set list of assumable roles |
| set user audit flags |
| set user labels |
| set user password |
| set user profiles |
| set/get file audit flags |
| use all defined labels |

*Table B-74* Commands Defined for the User Security Profile

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /opt/SUNWadm/2.1/bin/dbmgr | none | none | none | none | sys_trans_label |
| /opt/SUNWadm/2.1/bin/profmgr | none | none | none | none | sys_trans_label win_dac_read |
| /opt/SUNWadm/2.1/bin/usermgr | none | none | none | none | sys_trans_label |

## *Profile Name: inetd*

Definition: Privileges for programs executed by the inetd. Do not assign this profile to users.

*Table B-75* Actions Defined for the inetd Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-76* Authorizations Defined for the inetd Profile

| Authorizations |
|---|
| none |

*Table B-77* Commands Defined for the inetd Profile *(1 of 3)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/dt/bin/rpc.cmsd | none | none | none | none | 28<br>29<br>file_chown<br>file_dac_read<br>file_dac_write<br>file_mac_read<br>file_mac_write<br>file_nofloat<br>file_owner<br>file_setdac<br>file_setid<br>net_broadcast<br>net_downgrade_il<br>net_downgrade_sl<br>net_mac_read<br>net_nofloat<br>net_privaddr<br>net_reply_equal<br>proc_mac_read<br>proc_mac_write<br>proc_owner<br>proc_setid |

*Table B-77* Commands Defined for the inetd Profile *(2 of 3)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/dt/bin/rpc.ttdbserverd | none | none | none | none | 28<br>29<br>file_chown<br>file_dac_read<br>file_dac_write<br>file_mac_read<br>file_mac_write<br>file_nofloat<br>file_owner<br>file_setdac<br>file_setid<br>net_broadcast<br>net_downgrade_il<br>net_downgrade_sl<br>net_mac_read<br>net_nofloat<br>net_privaddr<br>net_reply_equal<br>proc_mac_read<br>proc_mac_write<br>proc_owner |
| /usr/sbin/in.ftpd | none | none | none | none | file_mac_write<br>net_privaddr<br>proc_audit_tcb<br>proc_setid |
| /usr/sbin/in.rexecd | none | none | none | none | net_privaddr<br>proc_audit_tcb<br>proc_setid |
| /usr/sbin/in.rlogind | none | none | none | none | file_chown<br>file_mac_write<br>file_setdac<br>net_privaddr<br>proc_audit_tcb |
| /usr/sbin/in.rshd | none | none | none | none | net_privaddr<br>proc_audit_tcb<br>proc_setid |

*Table B-77* Commands Defined for the inetd Profile *(3 of 3)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/in.telnetd | none | none | none | none | file_chown<br>file_mac_write<br>file_setdac<br>net_privaddr<br>proc_audit_tcb |
| /usr/sbin/in.tftpd | none | none | none | none | proc_chroot<br>proc_setid |
| /usr/sbin/sadmind | none | none | none | none | all |

## *Profile Name: boot*

Definition: System start-up/shutdown profile

*Table B-78* Actions Defined for the boot Profile

| Action(s) | Minimum label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| none | none | none | none | none | none |

*Table B-79* Authorizations Defined for the boot Profile

| Authorizations |
| --- |
| act as file owner |
| allocate device |
| bypass view of file contents on drag and drop |
| change file owner |
| downgrade file sensitivity label |
| enable logins |
| paste to a downgraded window |
| paste to an upgraded window |
| permit self-modification |
| print a PostScript file |
| print without label |
| remote login |
| set application search path |
| set attributes related to home directories |
| terminal login |
| set/get file audit flags |
| set file privileges |
| set idle time |
| set list of assumable roles |
| set user audit flags |
| set user identity |
| set user labels |
| set user password |
| set user profiles |
| set/get file audit flags |
| terminal login |
| upgrade file sensitivity label |
| use all defined labels |

*Table B-80* Commands Defined for the boot Profile *(1 of 3)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /sbin/mountall | none | ADMIN_HIGH | none | none | all |
| /usr/lib/lp/lpsched | ADMIN_HIGH | ADMIN_HIGH | none | none | file_chown<br>file_dac_read<br>file_dac_search<br>file_dac_write<br>file_downgrade_sl<br>file_mac_read<br>file_mac_search<br>file_mac_write<br>file_owner<br>file_setdac<br>file_setid<br>file_upgrade_sl<br>net_downgrade_sl<br>net_mac_read<br>net_setid<br>net_setpriv<br>net_upgrade_il<br>proc_audit_tcb<br>proc_mac_write<br>proc_nofloat<br>proc_owner<br>proc_setclr<br>proc_setid<br>proc_setsl<br>sys_trans_label |
| /usr/sbin/auditd | none | none | none | none | file_mac_write<br>proc_setclr<br>proc_setil<br>proc_setsl<br>sys_audit |

*Table B-80* Commands Defined for the boot Profile *(2 of 3)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/cron | ADMIN_LOW | ADMIN_HIGH | 0 | none | file_dac_read<br>file_mac_write<br>file_owner<br>net_mac_read<br>proc_audit_tcb<br>proc_nofloat<br>proc_setclr<br>proc_setid<br>proc_setil<br>proc_setsl<br>sys_audit |
| /usr/sbin/in.named | ADMIN_LOW | ADMIN_HIGH | 0 | none | file_mac_read<br>net_mac_read<br>net_privaddr<br>net_upgrade_sl<br>proc_dumpcore<br>proc_nofloat<br>proc_setclr<br>sys_net_config<br>sys_trans_label |
| /usr/sbin/inetd | ADMIN_LOW | ADMIN_HIGH | none | none | all |
| /usr/sbin/nscd | ADMIN_LOW | ADMIN_HIGH | none | none | file_dac_write<br>file_setid<br>net_mac_read<br>net_upgrade_sl<br>proc_dumpcore<br>proc_nofloat<br>proc_setclr<br>sys_net_config<br>sys_trans_label |
| /usr/sbin/rpcbind | ADMIN_HIGH | ADMIN_HIGH | none | none | all |

*Table B-80* Commands Defined for the boot Profile *(3 of 3)*

| Pathname | Minimum Label | Maximum Label | UID | GID | Privilege(s) |
|---|---|---|---|---|---|
| /usr/sbin/syslogd | ADMIN_LOW | ADMIN_HIGH | none | none | file_dac_write<br>file_mac_write<br>file_upgrade_il<br>net_downgrade_il<br>net_downgrade_sl<br>net_mac_read<br>net_privaddr<br>net_upgrade_il<br>net_upgrade_sl<br>proc_nofloat<br>proc_setclr<br>proc_setil<br>proc_setsl<br>sys_trans_label |
| /usr/sbin/tnd | none | none | none | none | net_downgrade_sl<br>net_mac_read<br>net_privaddr<br>proc_setclr<br>proc_setsl<br>sys_net_config |
| /usr/sbin | none | none | none | none | all |

# B

# *Index*

## A


accept command
  modified for Trusted Solaris, 382
  default profile, 493
access
  review of concepts, 305 to 317
Access Control Lists, *See* ACLs
access policy
  files, directories and file systems, 306
accounts
  assigning account type, 105
  assigning clearances, 116
  assigning comments, 103
  assigning execution profiles, 120 to 121
  assigning home directories, 113 to 115
  assigning idle time, 122 to 123
  assigning labels, 116 to 119
  assigning passwords, 106 to 113
  assigning roles, 122
  assigning shells, 104
  assigning user information, 102
  authorizations for setting up, 62
  configuring, *See* User Manager
  division of setup tasks, 61 to 62
  managing, 57 to 85
  planning, 58 to 60
  preconditions for setup, 58
  procedures for setting up, 124 to 155
  startup files, 65 to 85
  worksheet, 192
accreditation ranges
  *See also* label ranges
  described, 308
ACLs
  described, 305
actions
  adding new administrative, 24 to 26
  assigning to execution profiles, 221
  defined, 441
  in execution profiles, 195
  restricted by account profiles, 444
  using actions, 466
adb command
  default profile, 493
add_drv command
  default profile, 493
adjunct file
  *See also* `vfstab_adjunct` file
Admin Edit action, 12
admin role, *See* system administrator
ADMIN_HIGH label
  replacing with CIPSO label, 296
ADMIN_LOW label

installing publicly available software, 437
administrative actions
creating, 24 to 26
introduced, 444
administrative roles
applications
launching, 23
location on the desktop, 23
assuming
background, 4 to 5
procedure, 12 to 20
changing workspace SLs, 21
described, 89 to 90
password, 5
switching workspaces, 21
workspaces
display when role is assumed, 5
using, 7 to 24
adminvi command
default profile, 493
described, 12
adorned pathnames
described, 309
adornments
described, 44
aliases
setting up, 164, 167
All execution profile
allowing standard UNIX access to commands and actions, 442
ordering in an account's profile list, 443
allocatable devices
managing allocation, 420
allocate command
default profile, 493
allocate error state, 408, 413
allowed privileges, 440
APIs
*See also* privileges
Application Manager
as trusted process, 444
passing inheritable privileges, 454
using AdminSuite, 8
using Solstice_Apps, 8
using System_Admin, 8
applications
administrative
launching, 23
assigning forced privileges, 476
importing, 438
setting up with a real UID of 0, 473
setting up with effective UID of 0, 473
aspppd command
default profile, 493
at command
administrative differences, 73 to 82
running privileged commands, 75
Trusted Solaris changes, 81
at.allow file, 76
at.deny file, 76
atohexlabel command
default profile, 493
atq command
administrative differences, 73 to 82
Trusted Solaris changes, 81
atrm command
administrative differences, 73 to 82
Trusted Solaris changes, 82
attributes
*See also* security attributes
audio coprocessor, 409
audit command
default profile, 493
audit IDs
purpose when a role is assumed, 5
auditconfig command
default profile, 493
auditd command
default profile, 493
auditing
*See also* application auditing
auditreduce command
default profile, 494
auditstat command
default profile, 494

AUTH_*authorization name, See* authorizations and `auth_desc` file
`auth_name` file, 35
`auth_names.h` file, 33
authorizations
*See also* auth_desc file
*See also* privileges
adding, 33
administering printing, 383
assigning to execution profiles, 219
enable logins, 6
for specifying fields in the User Manager, 61
modify at admin, 78
modify at user, 79
modify cron admin, 79
modify cron user, 79
needed for users who assume secadmin role, 5
print without banners, 383
print without labels, 383
procedure for adding, 36
required for account setup, 62
required for User Manager, 92 to 94
self-modification, 94
`tsolprof` file, 32
autopush command
default profile, 494
awk command
default profile, 494

## B

banner pages
printing without, 397
banners on print jobs, 378
batch command
administrative differences, 73 to 82
running privileged commands, 75
body pages
labels, 368
specifying SLs to print instead of ILs, 396
boot execution profile, 457
bsmconv command
default profile, 494
bsmunconv command
default profile, 494

## C

cachefs, 327
cachefslog command
default profile, 494
cancel command
modified for Trusted Solaris, 382
default profile, 494
cat command
default profile, 494
cd command
default profile, 494
CDE actions, *See* actions
CDE configuration file default, 114
CDROM devices
`device_clean` script, 408
mounting, 343
changes to printing in Trusted Solaris, 364 to 393
chgrp command
default profile, 494
chk_encodings command
default profile, 494
chkey command
default profile, 494
chmod command
default profile, 494
chown command
default profile, 494
CIPSO
(Common Internet Protocol Security Option)
host type
described, 265
replacing ADMIN_HIGH, 296
use in packets, 250
classifications

described, 309
clear command
default profile, 494
clearances
*See also* labels
assigning, 116
described, 309
clri command
default profile, 494, 495
`.cm.rc` file, 68
cmp command
default profile, 495
CMW labels
*See also* labels
described, 309
on removable media, 403
col command
default profile, 495
commands
assigning to execution profiles, 217
privileged
run by cron(1MTSOL), 75
privileges, 461
trusted, 461
comments
assigning to accounts, 103
commercial applications
adding, 461
compartments
described, 309
compress command
default profile, 495
configuration files, *See* databases
`.copy_files` files
using, 70
cp command
default profile, 495
cpio command
default profile, 495
crash command
default profile, 495
cron command
administrative differences, 73 to 82
running privileged commands, 75
default profile, 495
cron.allow file, 76
cron.deny file, 76
cron command
Trusted Solaris changes, 80
crontab command
administrative differences, 73 to 82
csh command
default profile, 495
customizations
changing printer output, 372
cut command
default profile, 495

## D

DAC
(discretionary access control)
*See also* security policy
cautions about override
privileges, 460
described, 310
override privileges, 439
data objects, *See* objects
data packets, *See* packets
Database Manager
accessing network databases, 280
adding a host to a running
system, 348
databases
accessing `tnidb`(4TSOL), 254
accessing `tnrhdb` file, 254
accessing `tnrhtp`(4TSOL), 254
date command
default profile, 495
dbmgr command
default profile, 495
deallocate command
default profile, 495
default configuration, 413
default shells
assigning to accounts, 104

/dev/kmem kernel image file
security violation, 463
developers
responsibilities, 464
device allocation
lock files, 412
meeting the object reuse requirement, 403
device_allocate file
default contents, 414
device_clean scripts
for tape devices, 408
meeting object reuse requirements, 403
review, 407
devices
allocatable
default configuration, 413
associated security risks, 402
clearing prior to reuse, 403
frame buffer, 413
managing allocation, 420
non-allocatable
setting the label range, 413
policy for access to, 306
printers
default configuration, 413
required location of device special files, 407
setting policy for, 419
with removable media
special object reuse requirements, 403
df command
default profile, 495
diff command
default profile, 495
diff3 command
default profile, 495
dircmp command
default profile, 495
directories
changing flags, 321
changing labels and privileges, 336
security attributes, 318 to 326
sharing, 343
dirname command
default profile, 495
disable command
modified for Trusted Solaris, 382
default profile, 495
discretionary access control, *See* DAC
documents, policy for access to, 306
dominance of labels
described, 310
drvconfig command
default profile, 495
dtwm(1) command, 444
du command
default profile, 495

## E

echo command
default profile, 495
editing privileged executables, 472
eeprom command
default profile, 495
effective GIDs, *See* GIDs
effective UIDs, *See* UIDs
email
aliases, 164, 167
listing mail queues, 167
managing, 157 to 189
options, 176 to 179
overview, 158 to 164
switching mail tools, 180 to 189
troubleshooting, 171 to 176
enable command
modified for Trusted Solaris, 382
default profile, 496
enable logins authorization, 6
enabling logins
after a reboot, 5
env command
default profile, 496
ERRORS

section on man pages, 439
/etc/cron.d/CRON, cron(1MTSOL) lock file, 76
/etc/init.d directory
changing scripts, 457
starting and stopping scripts, 459
/etc/skel file, 69, 114
executable files
assigning forced privileges, 476
editing while preserving privileges, 472
privilege sets, 440
execution profiles
All
allowing standard UNIX access to commands and actions, 442
contents, 488
ordering in an account's profile list, 443
All Authorizations
contents, 488
assigning, 120, 121
Audit Control
contents, 488
Audit Review
contents, 488
Basic Actions
contents, 488
Basic Commands
contents, 489
boot, 457
contents, 492
controlling the use of actions, 444
Convenient Authorizations
contents, 489
creating new for boot commands, 457
described, 310
editing, 198, 232
Enable Login
contents, 489
importance of order, 442
inetd profile
contents, 492
Maintenance and Repair
contents, 489
managing, 193 to 232
Media Backup
contents, 489
Media Restore
contents, 489
NIS+ Administration
contents, 490
NIS+ Security Administration
contents, 490
Object Access Management
contents, 490
Object Label Management
contents, 490
Object Privilege Management
contents, 490
Outside Accred
contents, 490
overview, 194 to 197
Privileged Shells
contents, 490
using to start commands from run control scripts, 459
System Management
contents, 491
System Security
contents, 491
table of commands, 493 to 506
table of profile contents, 488 to 492
User Management
contents, 491
User Security
contents, 491
execve system call
inheriting privileges across, 450
replacing the executing program, 445
exporting software, 437
expr command
default profile, 496

## F

fallback mechanism
creating, 287
networks, 274

false command
default profile, 496
FDFS
mounting in Trusted Solaris, 328
fgrep command
default profile, 496
file command
default profile, 496
identifying multilevel directorie, 50
File Manager
as trusted process, 444
passing inheritable privileges, 454
Privileges dialog box, 453, 454
file privilege sets, 440
file systems
changing security attributes using mount -S command, 340
changing security attributes using newsecfs(1MTSOL), 339
changing security attributes using setfsattr(1MTSOL), 339
changing security attributes using `vfstab` file, 340
managing, 303, 344
review of concepts, 305 to 317
security attributes, 318 to 326
single label, 325
types, 327
file_mac_write privilege
resulting in a file's dominating its directory's SL, 356
file_upgrade_sl privilege
resulting in upgraded names, 356
files
changing flags, 321
changing labels, 320
changing labels and privileges, 336
changing privileges, 320
managing, 303 to 344
policy for access to, 306
review of concepts, 305 to 317
floating, *See* ILs
floppy devices
`device_clean` script, 408
fold command
default profile, 496
forced privileges, 440
fork system call
creating processes, 445
inheriting privileges across, 450
format command
default profile, 496
Front Panel
as trusted process, 444
modifying, 467
passing inheritable privileges, 454
Subpanels
as trusted processes, 444
fsck command
default profile, 496
fsdb command
default profile, 496
fsirand command
default profile, 496

## G

gateways
concepts, 240, 246
default, 271
per network, 271
getdents system call
restricting from returning upgraded names, 356
getfacl command
default profile, 496
getfattrflag command
default profile, 496
described, 321
identifying multilevel directories, 50
getfpriv command
default profile, 496
using to save privileges, 483
getfsattr command
default profile, 496
described, 324
getlabel command
default profile, 496

getmldadorn command
default profile, 496
getsldname command
default profile, 496
GIDs
(group IDs)
effective
alternative to privileges, 440
defaults, 460
defined, 438
in execution profiles, 195
grep command
default profile, 496
group IDs
assigning, 102
group names
assigning, 102
groupmgr command
default profile, 497
grpck command
default profile, 497

## H

halt command
default profile, 497
head command
default profile, 497
hexadecimal label equivalents
determining, 30
hextoalabel command
default profile, 497
Home dialog box
User Manager, 113 to 115
Host Manager
adding a host to a running system, 348
host types
CIPSO, 265
MSIX, 263
networking, 255 to 270
RIPSO, 267
sun_tsol, 257
TSIX, 260
unlabeled, 269
hostid command
default profile, 497
hostname command
default profile, 497
hosts
assigning templates, 282 to 286
networking concepts, 235, 246
worksheet, 485
HSFS
mounting in Trusted Solaris, 328

## I

icons
visibility
in the File Manager, 444
in the Workspace Menu, 444
id command
default profile, 497
identification and authentication
before assuming a role, 5
Identity dialog box
User Manager, 102
Idle dialog box
User Manager, 122, 123
IDs
*See also* audit IDs
*See also* GIDs
*See also* UIDs
ILs
(information labels)
*See also* labels
described, 310
displaying and hiding, 117 to 119
floating
described, 311
in.ftpd command
default profile, 497
in.named command
default profiles, 497
in.rexecd command
default profile, 497
in.rlogind command

default profile, 497
in.rshd command
default profile, 497
in.telnetd command
default profile, 497
in.tftpd command
default profile, 497
INET Domain sockets, *See* networking
inetd command
default profile, 497
information labels, *See* ILs
inheritable privileges, 450
init command
default profile, 497
`init.d` directory
starting commands during boot, 457
initialization files
Trusted Solaris differences, 65
installation
adding software, 463
internationalization
changing printer output, 372
interprocess communication, *See* IPC
IP Options field
use, 249

## J

join command
default profile, 497

## K

kernel switches
configurable, 354
kill command
default profile, 497
kmem(7D) kernel image file, 463
ksh command
default profile, 497

## L

label ranges
*See also* accreditation ranges
described, 311
devices, 414
setting on individual computers, 413
`label_encodings` file
distributing changes, 361, 362
procedures
printing without banners and trailers, 397
printing without labeled pages, 379, 399
printing without page labels, 398
labels
*See also* clearances
*See also* information labels
*See also* SLs
changing on files and directories, 336
described, 311
displaying, 117 to 119
distributing changes, 361 to 362
on imported and exported data, 411
physical, on removable media, 403
printed body pages, 368
Labels dialog box
User Manager, 116, 119
ldd command
default profile, 497, 498
libt6(3NTSOL), 76
`.link_files` file
using, 70
list_devices command
default profile, 498
ln command
default profile, 498
`local.login` file
defining printers, 399
lock files
for allocatable devices, 412
LOFS
mounting in Trusted Solaris, 328
log files
security violation from sharing, 463
`.login file`, 114

login
by administrative roles, 4 to 5
login sequence, 6
look command
default profile, 498
lp command
default profile, 498
lpadmin command
modified for Trusted Solaris, 382
default profile, 498
lpc command
modified for Trusted Solaris, 382
lpfilter command
default profile, 498
lpmove command
modified for Trusted Solaris, 382
default profile, 498
lpr command
modified for Trusted Solaris, 382
lprm command
modified for Trusted Solaris, 382
lpsched command
modified for Trusted Solaris, 382
default profile, 498
lpshut command
modified for Trusted Solaris, 382
default profile, 498
lpstat command
modified for Trusted Solaris, 382
default profile, 498
lpsystem command
modified for Trusted Solaris, 382
default profile, 498
lptest command
modified for Trusted Solaris, 382
lpusers command
default profile, 498
ls command
default profile, 498

## M

MAC
(mandatory access control)
bypassing restrictions, 439
cautions about override privileges, 462
described, 311
incoming packets, 253
outgoing packets, 252
override privileges, 439
mail
aliases, 164 to 167
listing mail queues, 167
managing, 157 to 189
messages, policy for access to, 306
options, 176 to 179
overview, 158 to 164
switching mail tools, 180 to 189
troubleshooting, 171 to 176
mailq command
default profile, 498
`.mailrc` file, 68
man command
default profile, 498
man pages
ERRORS sections, 439
mandatory access control, *See* MAC
markings
described, 312
minimum labels
assigning, 116
minimum SLs
described, 312
mkdir command
default profile, 498
default profiles, 498
mkfile command
default profile, 498
mkfs command
default profile, 498, 499
mldpwd command
default profile, 499
mldrealpath command
default profile, 499
identifying multilevel directories, 50
MLDs

*See also* SLDs
backing up, 51
described, 313
identifying, 50
working with, 43 to 53
modify at admin authorization, 78
modify at user authorization, 79
modify cron admin authorization, 79
modify cron user authorization, 79
more command
default profile, 499
mount command
nodevices option
controlling location of devices, 407
default profile, 499
mountall command
default profile, 499
mounts
managing, 303 to 344
permitted file systems, 327
procedure for TMPFS, 343
troubleshooting, 344
MSIX host type
described, 263
mt command
default profile, 499
multilevel directories, *See* MLDs
mv command
default profile, 499

## N

ncheck command
default profile, 499
`.netscape` file, 68
netstat command
default profile, 499
network accreditation range
example, 279
network accreditation ranges
described, 237
requirements, 239
setting up, 290
network interfaces
configuring, 291 to 296
described, 237
requirements, 239
networking
concepts, 235 to 246
networks
default labeling, 253 to 279
fallback mechanism, 274
procedures for configuring, 280 to 302
security attributes, 248 to 279
using templates, 273
newaliases command
default profile, 499
newfs command
default profile, 499
newkey command
default profile, 499
newsecfs command
default profile, 499
newsecfs command)
described, 324
`.newsrc` file, 68
NFS
mounting in Trusted Solaris, 328
nice command
default profile, 499
NIS+
configuration files not administered, 347
tables
adding protected data, 348
new in Trusted Solaris, 347
NIS+ client host
adding to a running system, 348
NIS+, managing, 345 to 348
nisaddcred command
default profile, 499
nisaddent command
default profile, 499
niscat command

default profile, 499
nischgrp command
default profile, 499
nischmod command
default profile, 500
nischown command
default profile, 500
nischttl command
default profile, 500
nisclient command
default profile, 500
nisctl command
default profile, 500
nisdefaults command
default profile, 500
niserror command
default profile, 500
nisgrep command
default profile, 500
nisgrpadm command
default profile, 500
nisinit command
default profile, 500
nisln command
default profile, 500
nislog command
default profile, 500
nismatch command
default profile, 500
nismkdir command
default profile, 500
nispasswd command
default profile, 500
nisping command
default profile, 500
nispopulate command
default profile, 500
nisrm command
default profile, 500
nisrmdir command
default profile, 500
nisserver command
default profile, 500
nissetup command
default profile, 500
nisshowcache command
default profile, 500
nisstat command
default profile, 500
nistbladm command
default profile, 500
nistest command
default profile, 500
nistnsetup command
default profile, 500
nistntime command
default profile, 500
nisupdkeys command
default profile, 501
non-administrative roles
described, 88, 89
nroff command
default profile, 501
nscd command
default profiles, 501

## O

object reuse
how device allocation satisfies, 403
object-reuse
review, 403
oper role, *See* system operator

## P

packets
IP options, 249
security attributes, 249 to 254
page command
default profile, 501
PAM
*See pluggable authentication module*, 111
`pam_headers.h` file
settings for password values, 110
passwords

assigning, 106, 113
requiring changes, 111
role, 5
rules for manual creation, 109
pattr command
default profile, 501
PCFS
mounting in Trusted Solaris, 328
pclear command
default profile, 501
pcred command
default profile, 501
permission bits
described, 314
permissions
described, 305
pfiles command
default profile, 501
pflags command
default profile, 501
pfsh command
default profile, 501
purpose, 11
pfsh command, *See also* profile shell
pg command
default profile, 501
ping command
default profile, 501
pkgadd command
default profile, 501
pkgchk command
default profile, 501
pkginfo command
default profile, 501
pkgrm command
default profile, 501
plabel command
default profile, 501
pldd command
default profile, 501
pluggable authentication module
default settings, 111
pmap command
default profile, 501
policy, *See* security policy
porting software, 438
poweroff command
default profile, 501
pr command
default profile, 501
praudit command
default profile, 501
principle of least privilege, 440
Printer Manager
launching, 384
PRINTER variable
printing without labels, 399
printers
label ranges
setting, 413
printing
authorizations, 383
configuring labels and text, 372
specifying SLs to print instead of ILs on body pages, 396
suppressing page labels, 379
without banners and trailers, 397
without banners authorization, 383
without page labels, 398
`priv_name` file, 39
`priv_names.h` file, 37
PRIV_*privilege name*, *See* privileges and `priv_desc` file
privilege debugging
setting tsol_privs_debug, 357
privilege sets
process, 445
privileged commands
run by cron command(1MTSOL), 75
privileged programs, 461
privileges
*See also* authorizations
*See also* `priv_desc` file
adding, 37
allowed, 454
alternatives to using, 440

bracketing, 440
changing on files and directories, 336
DAC override, 439
defined, 438
described, 314
effective set
defined, 445
example, 441
in execution profiles, 196
file_dac_read
as an override privilege, 439
file_dac_write
as an override privilege, 439
file_mac_read
as an override privilege, 439
file_mac_write
as an override privilege, 439
forced
assigning, 453, 476
inheritable, 450, 454
inheritable set
defined, 445
inheritance, 442
MAC override, 439
making available to commands, 453
non-obvious reasons for
requiring, 463
override, 96
defined, 439
permitted set
defined, 445
principle of least privilege, 440
procedure for adding, 40
required, 95, 439
running with all, 444
saved set
defined, 445
saving and restoring an edited
executable, 483
used by commands in the pfsh
shell, 443
process privilege sets, 445
processes
described, 314
relationship to programs, 445
PROCFS
mounting in Trusted Solaris, 329
procplabel command
default profile, 502
procppriv command
default profile, 502
procpprivtest command
default profile, 502
Profile dialog box
User Manager, 120, 121
Profile Manager
assigning actions, 221
assigning authorizations, 219
assigning commands, 217
filtering profiles, 200
procedures for using, 223 to 232
setting label ranges, 215
specifying privileges for commands
and actions, 454
using, 198 to 232
profile shell
effect on commands, 442
managing inherited privileges, 443
purpose, 11
startup algorithm, 67
profiles
*See* execution profiles
profmgr command
default profile, 502
programs
commercial
assigning privileges to, 453
new, trusted
assigning privileges to, 453
relationship to processes, 445
trusted
defined, 461
trustworthy
defined, 461
prtconf command
default profile, 502
prun command
default profile, 502
ps command

default profile, 502
psig command
default profile, 502
pstack command
default profile, 502
pstp command
default profile, 502
ptime command
default profile, 502
ptree command
default profile, 502
pwait command
default profile, 502
pwck command
default profile, 502
pwd command
default profile, 502
pwdx command
default profile, 502

## R

rc scripts
shell use, 443
rc2.d directory
S05RMTMPFILES script, 6
rcp command
default profile, 502
required privilege, 462
rdist command
default profile, 502
real UID
root
required for applications, 464
root requirement for installation, 463
reboot
preventing disable of logins, 26
re-enabling logins, 5
reboot command
default profile, 502
reject command
modified for Trusted Solaris, 382
default profile, 502
rem_drv command
default profile, 502
renice command
default profile, 502
RIPSO
(Revised Internet Protocol Security Option)
host type
described, 267
use in packets, 251
rlogin command
default profile, 502
rm command
default profile, 502, 503
rmdir command
default profile, 503
roles
administrative, 89 to 90
combining administrative roles, 94
contrasted with user accounts, 88
creating, 95 to 98
override privileges, 96
required privileges, 95
managing, 87 to 98
non-administrative, 88 to 89
password, 5
procedure for creating, 98
root, 89
security administrator contrasted with system administrator, 91
Roles dialog box
User Manager, 122
root
saving and restoring NIS+ tables, 349
root role
compared with privileges, 439
using to install applications, 464
root UID
required for applications, 464
requirement for installation, 463
routing
assigning default, 297
concepts, 240 to 246

procedure for setup, 297
setting up, 271 to 302
rpc.cmsd command
default profile, 503
rpc.ttdbserverd command
default profile, 503
rpcbind command
default profile, 503
rsh command
default profile, 503
run control scripts
modifying, 457
shell use, 443
runpd command
default profile, 503
dependency on tsol_priv_debug setting, 357
rup command
default profile, 503

## S

sadmind command
default profile, 503
SAMP
(Security Attribute Modulation Protocol)
SATMP
(Security Attribute Token Mapping Protocol)
/sbin/rc*n* scripts
Trusted Solaris modifications, 458
/sbin/sysh shell
using during boot, 458
script command
default profile, 503
sdiff command
default profile, 503
secadmin role, *See* security administrator
security administrators
accessing the Printer Manager, 384
account management responsibilities, 61
contrasted with system administrators, 91
described, 314
enable logins authorization requirement, 5
modifying window configuration files, 467
system integrity
assessing after reboots, 5
security attributes
described, 315
file systems, 318 to 326
security domains
multiple described, 238
single described, 236
security features
identification and authentication for roles, 5
security mechanisms
extendable by security administrator, 31
security policy
described, 315
principle of least privilege, 440
sed command
default profile, 503
self-modification authorization, 94
sendmail command
using, 168 to 171
sensitivity labels, *See* SLs
session clearances
described, 316
setfacl command
default profile, 503
setfattrflag command
default profile, 503
described, 321
setfpriv command
restricting assignment of forced and allowed privileges, 454
setfpriv command, 453
default profile, 503
setfsattr command

default profile, 503
described, 325
setlabel command
default profile, 503
sh command
default profile, 503
share command
default profile, 503
shareall command
default profile, 503
sharing directories, 343
shell scripts
summary of Trusted Solaris behavior, 470
writing privileged, 479
writing privileged using standard shells, 480
shells
assigning to accounts, 104
sysh(1MTSOL), 458
showmount command
default profile, 503
single-level directories, *See* SLDs
skeleton directories
defining printers, 399
use in Trusted Solaris, 69
skeleton path, 114
SLDs
described, 316
working with, 43 to 53
SLDs (single-label directories)
*See also* MLDs
sleep command
default profile, 503
SLs
(sensitivity labels)
*See also* labels
described, 316
displaying and hiding, 117 to 119
snoop command
default profile, 504
software
exporting multiple SLs, 437
importing, 438
multiple SLs, 437
installing publicly available software at ADMIN_LOW, 437
porting
reasons against, 463
sort command
default profile, 504
spell command
default profile, 504
spray command
default profile, 504
spreadsheet, policy for access to, 306
standard UNIX shell
effect on commands, 442
startup files
`.cm.rc` file, 68
configuring accounts, 65 to 85
`.mailrc` file, 68
`.netscape` file, 68
`.newsrc` file, 68
procedures for customizing, 83 to 85
`rc2 .d/S05RMTMPFILES`, 6
read at window system startup, 65
strace command
default profile, 504
strict dominance
described, 316
stty command
default profile, 504
sun_tsol host type
described, 257
swap command
default profile, 504
swmtool command
default profile, 504
sysh shell
using during boot, 458
trusted processes, 443
syslogd command
default profile, 504
system accreditation range
described, 317

system administrator
account management
responsibilities, 61
adding a host, 348
contrasted with security
administrator, 91
system security
violations, 463
system shell
described, 443
System V interprocess communication, *See* System V IPC
System_Admin folder
using administrative actions, 444

## T

tail command
default profile, 504
tape devices
`device_clean` scripts, 408
tar command
default profile, 504
allocating a media device, 411
tbl command
default profile, 504
TCP address
(Transmission Control Protocol address)
templates
assigning to hosts, 282, 286
test command
default profile, 504
testfpriv command
default profile, 504
tfind command
default profile, 504
time command
default profile, 504
TMPFS
mounting in Trusted Solaris, 329
mounting procedure, 343
tnchkdb command
default profile, 504
tnctl command
default profile, 504
tnd command
default profile, 504
`tnidb` file
accessing, 254
tninfo command
default profile, 504
`tnrhdb` file
accessing, 254
`tnrhtp` file
accessing, 254
tokmapctl command
default profile, 505
tokmapd command
default profile, 505
touch command
default profile, 505
troff command
default profile, 505
true command
default profile, 505
truss command
default profile, 505
Trusted Computing Base, *See* TCB
trusted networks
*See* networks
trusted path attribute
tools requiring, 90
Trusted Path menu, 5
trusted processes
defined, 444
launching actions, 444
passing inheritable privileges, 454
trusted programs, 461
adding, 464
Trusted Solaris
changes to printing, 364 to 393
trusted stripe
changing display, 42
trustworthy programs, 461
TSIX host type

described, 260
*tsol_attr* flag
described, 323
tsol_enable configurable kernel switch, 355
tsol_enable_il_floating configurable kernel switch, 355
tsol_hide_upgraded_names configurable kernel switch
defined, 356
tsol_privs_debug configurable kernel switch
defined, 357
tsol_privs_debug switch
described, 460
tsol_reset_il_on_exec configurable kernel switch
defined, 356
`tsol_separator.ps` file
procedures
specifying SLs to print instead of ILs, 396
tsolxagent command
default profile, 505
ttsession command
default profile, 505
tty command
default profile, 505
tunefs command
default profile, 505
two-person control
adding privileges to progams, 464

## U

UDP address
(User Datagram Protocol address)
UFS
mounting in Trusted Solaris, 329
ufsdump command
default profile, 505
ufsrestore command
default profile, 505
UIDs
(user IDs)
assigning, 102
effective
alternative to privileges, 440
defaults, 460
defined, 438
effective UID of root, 464
in execution profiles, 195
umount command
default profile, 505
umountall command
default profile, 505
uname command
default profile, 505
unbundled Sun applications
adding, 461
uncompress command
default profile, 505
uniq command
default profile, 505
UNIX domain socket
used by cron(1MTSOL) and its clients, 76
unlabeled host type
described, 269
unshare command
default profile, 505
unshareall command
default profile, 505
upgraded names
defined, 356
user accounts
*See also* accounts
*See also* User Manager
user accreditation range
described, 317
user clearances
described, 317
User Manager
assigning account type, 105
assigning passwords, 106 to 113

assigning profiles in the desired order, 442
assigning profiles to accounts, 454
data entries explained, 100 to 123
Home dialog box, 113 to 115
identity dialog box, 102
Idle dialog box, 122 to 123
Labels dialog box, 116 to 119
procedures, 124 to 155
Profile dialog box, 120 to 121
required authorizations, 92, 94
Roles dialog box, 122
worksheet, 192
user names
assigning, 102
usermgr command
default profile, 505
users
*See also* User Manager

## V

vmstat command
default profile, 505

## W

whereis command
default profile, 506
which command
default profile, 506
who command
default profile, 506
whoami command
default profile, 506
window manager, 444
window system
trusted processes, 444
passing inheritable privileges, 454
worksheets
for hosts, 485
for user accounts, 192
Workspace Menu
as trusted process, 444
modifying, 467
workspaces
administrative
using, 7 to 24
writeaudit command
default profile, 506

## X

xhost command
default profile, 506
`Xtsolusersession` script, 444