Solaris Trusted Extensions Administrator's Procedures

Beta

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–7309–10
January 2010

100121@23031

Contents

Figures

Tables

Preface

The *Solaris Trusted Extensions Administrator's Procedures* guide provides procedures for enabling and initially configuring Solaris™ Trusted Extensions on the Solaris Operating System. This guide also provides procedures for managing users, zones, devices, and hosts that are labeled with Solaris Trusted Extensions software.

Note – This Solaris release supports systems that use the SPARC® and x86 families of processor architectures. The supported systems appear in the Solaris OS: Hardware Compatibility Lists (`http://www.sun.com/bigadmin/hcl`). This document cites any implementation differences between the platform types.

In this document these x86 related terms mean the following:

- "x86" refers to the larger family of 64-bit and 32-bit x86 compatible products.
- "x64" relates specifically to 64-bit x86 compatible CPUs.
- "32-bit x86" points out specific 32-bit information about x86 based systems.

For supported systems, see the *Solaris OS: Hardware Compatibility Lists*.

Who Should Use This Guide

This guide is for knowledgeable system administrators and security administrators who are configuring and administering Trusted Extensions software. The level of trust that is required by your site security policy, and your level of expertise, determines who can perform the configuration tasks.

Administrators should be familiar with Solaris administration. In addition, administrators should understand the following:

- The security features of Trusted Extensions and your site security policy
- Basic concepts and procedures for using a host that is configured with Trusted Extensions, as described in the *Solaris Trusted Extensions User's Guide*
- How administrative tasks are divided among roles at your site

Trusted Extensions and the Solaris Operating System

Trusted Extensions runs on top of the Solaris Operating System (Solaris OS). Because Trusted Extensions software can modify the Solaris OS, Trusted Extensions can require specific settings for Solaris installation options. Part I of this guide describes how to prepare the Solaris OS for Trusted Extensions, how to enable Trusted Extensions, and how to initially configure the software. Part II of this guide describes how to administer the uniquely Trusted Extensions features of the system.

How the Solaris Trusted Extensions Guides Are Organized

The Solaris Trusted Extensions documentation set consists of the following guides.

Title of Guide	Topics	Audience
Solaris Trusted Extensions User's Guide	Describes the basic features of Solaris Trusted Extensions. This book contains a glossary.	End users, administrators, developers
Solaris Trusted Extensions Administrator's Procedures	For this release, Part I describes how to prepare for, enable, and initially configure Trusted Extensions. Part I replaces *Solaris Trusted Extensions Installation and Configuration*. Part II describes how to administer a Trusted Extensions system. This guide contains a glossary.	Administrators, developers
Solaris Trusted Extensions Developer's Guide	Describes how to develop applications with Solaris Trusted Extensions.	Developers, administrators
Solaris Trusted Extensions Label Administration	Provides information about how to specify label components in the label encodings file.	Administrators
Compartmented Mode Workstation Labeling: Encodings Format	Describes the syntax used in the label encodings file. The syntax enforces the various rules for well-formed labels for a system.	Administrators

Related Books From Sun Microsystems

The following guides contain information that is useful when you prepare for and run Solaris Trusted Extensions software.

Solaris Express Installation Guide: Planning for Installation and Upgrade – Provides guidance on installing the Solaris OS

Solaris Express Installation Guide: Custom JumpStart and Advanced Installations – Provides guidance on installation methods and configuration options

Book Title	Topics
System Administration Guide: Basic Administration	User accounts and groups, server and client support, shutting down and booting a system, managing services, and managing software (packages and patches)
System Administration Guide: Advanced Administration	Terminals and modems, system resources (disk quotas, accounting, and crontabs), system processes, and troubleshooting Solaris software problems
System Administration Guide: Devices and File Systems	Removable media, disks and devices, file systems, and backing up and restoring data
System Administration Guide: IP Services	TCP/IP network administration, IPv4 and IPv6 address administration, DHCP, IPsec, IKE, Solaris IP filter, Mobile IP, IP network multipathing (IPMP), and IPQoS
System Administration Guide: Network Interfaces and Network Virtualization	Networking stack, NIC driver property configuration, network interface configuration, administration of VLANs and link aggregations, configuring WiFi wireless networking.
System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)	DNS, NIS, and LDAP naming and directory services, including transitioning from NIS to LDAP and transitioning from NIS+ to LDAP
System Administration Guide: Network Services	Web cache servers, time-related services, network file systems (NFS and Autofs), mail, SLP, and PPP
System Administration Guide: Security Services	Auditing, device management, file security, BART, Kerberos services, PAM, Solaris Cryptographic Framework, privileges, RBAC, SASL, and Solaris Secure Shell
System Administration Guide: Virtualization Using the Solaris Operating System	Resource management features, which enable you to control how applications use available system resources; zones software partitioning technology, which virtualizes operating system services to create an isolated environment for running applications; and virtualization using Sun™ xVM hypervisor technology, which supports multiple operating system instances simultaneously
Solaris CIFS Administration Guide	Solaris CIFS service, which enables you to configure a Solaris system to make CIFS shares available to CIFS clients; and native identity mapping services, which enables you to map user and group identities between Solaris systems and Windows systems
Solaris ZFS Administration Guide	ZFS storage pool and file system creation and management, snapshots, clones, backups, using access control lists (ACLs) to protect ZFS files, using ZFS on a Solaris system with zones installed, emulated volumes, and troubleshooting and data recovery

Book Title	Topics
System Administration Guide: Solaris Printing	Solaris printing topics and tasks, using services, tools, protocols, and technologies to set up and administer printing services and printers

Books From Elsewhere

Your site security policy document – Describes the security policy and security procedures at your site

The administrator guide for your currently installed operating system – Describes how to back up system files

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites that are mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for shells that are included in the Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Solaris release.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

PART I

Initial Configuration of Trusted Extensions

This section describes how to prepare for running Trusted Extensions. The chapters cover enabling Trusted Extensions and initial setup. For networked systems, initial setup includes configuring LDAP for Trusted Extensions.

Chapter 1, "Security Planning for Trusted Extensions," describes the security issues that you need to consider when configuring Trusted Extensions software on one or more Solaris systems.

Chapter 2, "Configuration Roadmap for Trusted Extensions," contains task maps for adding Trusted Extensions software to Solaris systems.

Chapter 3, "Adding Solaris Trusted Extensions Software to the Solaris OS (Tasks)," provides instructions on preparing a Solaris system for Trusted Extensions software. It also includes instructions on enabling Trusted Extensions.

Chapter 4, "Configuring Trusted Extensions (Tasks)," provides instructions on configuring Trusted Extensions software on a system with a monitor.

Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)," provides instructions on configuring LDAP for Trusted Extensions.

Chapter 6, "Configuring a Headless System With Trusted Extensions (Tasks)," describes how to configure and administer Trusted Extensions software on a headless system.

CHAPTER 1

Security Planning for Trusted Extensions

Solaris™ Trusted Extensions implements a portion of your site's security policy in software. This chapter provides an overview of the security and administrative aspects of configuring the software.

Planning for Security in Trusted Extensions

This section outlines the planning that is required before enabling and configuring Trusted Extensions software.

For a checklist of Trusted Extensions configuration tasks, see Appendix B, "Configuration Checklist for Trusted Extensions." If you are interested in localizing your site, see "For International Customers of Trusted Extensions" on page 30. If you are interested in running an evaluated configuration, see "Understanding Your Site's Security Policy" on page 28.

Understanding Trusted Extensions

The enabling and configuration of Trusted Extensions involves more than loading executable files, specifying your site's data, and setting configuration variables. Considerable background knowledge is required. Trusted Extensions software provides a labeled environment that is based on the following concepts:

- Capabilities that in most UNIX® environments are assigned to superuser are available to discrete administrative roles.
- In addition to UNIX permissions, access to data is controlled by special security tags. These tags are called *labels*. Labels are assigned to users, processes, and objects, such as data files and directories.
- The ability to override security policy can be assigned to specific users and applications.

Understanding Your Site's Security Policy

Trusted Extensions effectively enables you to integrate your site's security policy with the Solaris OS. Thus, you need to have a good understanding of the scope of your policy and the ability of Trusted Extensions software to accommodate that policy. A well-planned configuration must provide a balance between consistency with your site security policy and convenience for users who are working on the system.

Trusted Extensions is configured by default to conform with the Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408) at Assurance Level EAL4 against the following protection profiles:

- Labeled Security Protection Profile
- Controlled Access Protection Profile
- Role-Based Access Control Protection Profile

To meet these evaluated levels, you must configure LDAP as the naming service. Note that your configuration might no longer conform with the evaluation if you do any of the following:

- Change the kernel switch settings in the `/etc/system` file.
- Turn off auditing or device allocation.
- Change the default entries in public files in the `/usr` directory.

For more information, see the Common Criteria web site (`http://www.commoncriteriaportal.org/`).

Devising an Administration Strategy for Trusted Extensions

The root user or the System Administrator role is responsible for enabling Solaris Trusted Extensions. You can create roles to divide administrative responsibilities among several functional areas:

- The security administrator is responsible for security-related tasks, such as setting up and assigning sensitivity labels, configuring auditing, and setting password policy.
- The system administrator is responsible for the non-security aspects of setup, maintenance, and general administration.
- The primary administrator is responsible for creating rights profile for the security administrator, and for fixing problems when the security and system administrators do not have sufficient privilege.
- More limited roles can be configured. For example, an operator could be responsible for backing up files.

As part of your administration strategy, you need to decide the following:

- Which users are handling which administration responsibilities
- Which non-administrative users are allowed to run trusted applications, meaning which users are permitted to override security policy, when necessary
- Which users have access to which groups of data

Devising a Label Strategy

Planning labels requires setting up a hierarchy of sensitivity levels and a categorization of information on your system. The label encodings file contains this type of information for your site. You can use one of the label_encodings files that are supplied on the Solaris Trusted Extensions installation media. You could also modify one of the supplied files, or create a new label_encodings file that is specific to your site. The file must include the Sun-specific local extensions, at least the COLOR NAMES section.

Caution – If you are supplying a label_encodings file, you must have the final version of the file ready before rebooting the system after you enable the Trusted Extensions service. The file should be on removable media.

Planning labels also involves planning the label configuration. After enabling the Trusted Extensions service, you need to decide if the system can run at a single label only, or if the system can run at multiple labels. If all of your non-administrative users can operate at the same security label, select a single-label system.

You can also configure whether labels display and which label name format is displayed. For more information, see *Solaris Trusted Extensions Label Administration*. You can also refer to *Compartmented Mode Workstation Labeling: Encodings Format*.

For International Customers of Trusted Extensions

When localizing a `label_encodings` file, international customers must localize the label names *only*. The administrative label names, `ADMIN_HIGH` and `ADMIN_LOW`, must not be localized. All labeled hosts that you contact, from any vendor, must have label names that match the label names in the `label_encodings` file.

Trusted Extensions supports fewer locales than does the Solaris OS. When you are working in a locale that Trusted Extensions does not support, text that is specific to Trusted Extensions, such as error messages about labels, is not translated into your locale. Solaris software continues to be translated into your locale.

Planning System Hardware and Capacity for Trusted Extensions

System hardware includes the system itself and its attached devices. Such devices include tape drives, microphones, CD-ROM drives, and disk packs. Hardware capacity includes system memory, network interfaces, and disk space.

- Follow the recommendations for installing a Solaris release, as described in *Solaris Express Installation Guide: Planning for Installation and Upgrade*. Trusted Extensions features can add to those requirements:

  Memory beyond the suggested minimum is required on the following systems:

  - Systems that run the Solaris Management Console, a required administrative GUI
  - Systems that run at more than one sensitivity label
  - Systems that are used by users who can assume an administrative role
- More disk space is required on the following systems:
  - Systems that store files at more than one label
  - Systems whose users can assume an administrative role

Planning Your Trusted Network

For assistance in planning network hardware, see Chapter 2, "Planning an IPv4 Addressing Scheme (Tasks)," in *System Administration Guide: IP Services*.

As in any client-server network, you need to identify hosts by their function, that is, server or client, and configure the software appropriately. For assistance in planning, see *Solaris Express Installation Guide: Custom JumpStart and Advanced Installations*.

Trusted Extensions software recognizes two host types, labeled and unlabeled. Each host type has a default security template, as shown in Table 1–1.

TABLE 1–1 Default Host Templates in Trusted Extensions

Host Type	Template Name	Purpose
`unlabeled`	`admin_low`	At initial boot, labels the global zone. After initial boot, identifies hosts that send unlabeled packets.
`cipso`	`cipso`	Identifies hosts or networks that send CIPSO packets. CIPSO packets are labeled.

If your network can be reached by other networks, you need to specify accessible domains and hosts. You also need to identify which Trusted Extensions hosts are going to serve as gateways. You need to identify the label accreditation range for these gateways, and the sensitivity label at which data from other hosts can be viewed.

The `smtnrhtp(1M)` man page provides a complete description of each host type with several examples.

Planning for Zones in Trusted Extensions

Trusted Extensions software is added to the Solaris OS in the global zone. You then configure non-global zones that are labeled. You can create one labeled zone for every unique label, though you do not need to create a zone for every label.

Part of zone configuration is configuring the network. Labeled zones must be configured to communicate with the global zone and with other zones on the network.

- The X server that runs the desktop display is available only from the global zone. In the Solaris OS, the loopback interface, `lo0`, can be used to communicate with the global zone. Therefore, the desktop display is available to non-global zones over `lo0`.
- By default, non-global zones use the global zone to reach the network. In the Solaris OS, each non-global zone can be configured with a unique default route that does not use the global zone.

Trusted Extensions Zones and Solaris Zones

Labeled zones differ from typical Solaris zones. Labeled zones are primarily used to segregate data. In Trusted Extensions, regular users cannot remotely log in to a labeled zone. The only interactive interface to a labeled zone is by using the zone console. Only root can gain access to the zone console.

Zone Creation in Trusted Extensions

To create a labeled zone involves copying the entire Solaris OS, and then starting the services for the Solaris OS in every zone. The process can be time-consuming. A faster process is to create one zone, then to clone the contents of that zone. The following table describes your options for zone creation in Trusted Extensions.

Zone Creation Method	Effort Required	Characteristics of This Method
Create each labeled zone from scratch.	Configure, initialize, install, customize, and boot each labeled zone.	■ This method is useful for creating one or two additional zones. The zones can be upgraded. ■ This method is time-consuming.
Create additional labeled zones from a ZFS snapshot of the first labeled zone.	Configure, initialize, install, and customize one zone. Use this zone as a ZFS snapshot for additional labeled zones.	■ This method is the fastest method. This method makes every zone a file system, and thus provides isolation.

Planning for Multilevel Access

Typically, printing and NFS are configured as multilevel services. To access multilevel services, a properly configured system requires that every zone be able to access one or more network addresses. The following configurations provide multilevel services:

- As in the Solaris OS, one IP address is assigned for every zone, including the global zone. A refinement of this configuration is to assign a separate network information card (NIC) to each zone. Such a configuration is used to physically separate the single-label networks that are associated with each NIC.
- One `all-zones` address is assigned. One or more zones can have zone-specific addresses.

A system that meets the following two conditions cannot provide multilevel services:

- One IP address is assigned that the global zone and the labeled zones share.
- No zone-specific addresses are assigned.

If users in labeled zones are not supposed to have access to a local multilevel printer, and you do not need NFS exports of home directories, then you can assign one IP address to a system that you configure with Trusted Extensions. On such a system, multilevel printing is not supported, and home directories cannot be shared. A typical use of this configuration is on a laptop.

Planning for the LDAP Naming Service in Trusted Extensions

If you are not planning to install a network of labeled systems, then you can skip this section.

If you plan to run Trusted Extensions on a network of systems, use LDAP as the naming service. For Trusted Extensions. a populated Sun Java™ System Directory Server (LDAP server) is required when you configure a network of systems. If your site has an existing LDAP server, you can populate the server with Trusted Extensions databases. To access the server, you set up an LDAP proxy on a Trusted Extensions system.

If your site does not have an existing LDAP server, you then plan to create an LDAP server on a system that is running Trusted Extensions software. The procedures are described in Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)."

Planning for Auditing in Trusted Extensions

By default, auditing is turned on when Trusted Extensions is installed. Therefore, by default, `root` login and `root` logout are audited. To audit the users who are configuring the system, you can create roles early in the configuration process. For the procedure, see "Creating Roles and Users in Trusted Extensions" on page 79.

Planning auditing in Trusted Extensions is the same as in the Solaris OS. For details, see Part VII, "Solaris Auditing," in *System Administration Guide: Security Services*. While Trusted Extensions adds classes, events, and audit tokens, the software does not change how auditing is administered. For Trusted Extensions additions to auditing, see Chapter 24, "Trusted Extensions Auditing (Overview)."

Planning User Security in Trusted Extensions

Trusted Extensions software provides reasonable security defaults for users. These security defaults are listed in the Table 1–2. Where two values are listed, the first value is the default. The security administrator can modify these defaults to reflect the site's security policy. After the security administrator sets the defaults, the system administrator can create all the users, who inherit the established defaults. For descriptions of the keywords and values for these defaults, see the `label_encodings(4)` and `policy.conf(4)` man pages.

TABLE 1–2 Trusted Extensions Security Defaults for User Accounts

File name	Keyword	Value
`/etc/security/policy.conf`	`IDLECMD`	`lock` | `logout`
	`IDLETIME`	`30`

TABLE 1–2 Trusted Extensions Security Defaults for User Accounts *(Continued)*

File name	Keyword	Value
	LABELVIEW	showsl | hidesl
	CRYPT_ALGORITHMS_ALLOW	1,2a,md5
	CRYPT_DEFAULT	_unix_
	LOCK_AFTER_RETRIES	no | yes
	PRIV_DEFAULT	basic
	PRIV_LIMIT	all
	AUTHS_GRANTED	solaris.device.cdrw
	PROFS_GRANTED	Basic Solaris User
LOCAL DEFINITIONS section of /etc/security/tsol/label_encodings	Default User Clearance	CNF NEED TO KNOW
	Default User Sensitivity Label	PUBLIC

The system administrator can set up a standard user template that sets appropriate system defaults for every user. For example, by default. each user's initial shell is a Bourne shell. The system administrator can set up a template that gives each user a C shell. For more information, see the Solaris Management Console online help for User Accounts.

Devising a Configuration Strategy for Trusted Extensions

Allowing the `root` user to configure Trusted Extensions software is not a secure strategy. The following describes the configuration strategy from the most secure strategy to the least secure strategy:

- A two-person team configures the software. The configuration process is audited.

  Two people are at the computer when the software is enabled. Early in the configuration process, this team creates local users and roles. The team also sets up auditing to audit events that are executed by roles. After roles are assigned to users, and the computer is rebooted, the software enforces task division by role. The audit trail provides a record of the configuration process. For an illustration of a secure configuration process, see Figure 1–1.

  Note – If site security requires separation of duty, a trusted administrator completes “Create Rights Profiles That Enforce Separation of Duty” on page 80 before creating users or roles. In this customized configuration, one role manages security, including users' security attributes. The other role manages the non-security attributes of systems and users.

- One person enables and configures the software by assuming the appropriate role. The configuration process is audited.

  Early in the configuration process, the `root` user creates a local user and roles. This user also sets up auditing to audit events that are executed by roles. Once roles have been assigned to the local user, and the computer is rebooted, the software enforces task division by role. The audit trail provides a record of the configuration process.
- One person enables and configures the software by assuming the appropriate role. The configuration process is not audited.

  By using this strategy, no record is kept of the configuration process.
- The `root` user enables and configures the software. The configuration process is audited.

  The team sets up auditing to audit every event that `root` performs during configuration. With this strategy, the team must determine which events to audit. The audit trail does not include the name of the user who is acting as `root`.
- The `root` user enables and configures the software.

Task division by role is shown in the following figure. The security administrator sets up auditing, protects file systems, sets device policy, determines which programs require privilege to run, and protects users, among other tasks. The system administrator shares and mounts file systems, installs software packages, and creates users, among other tasks.

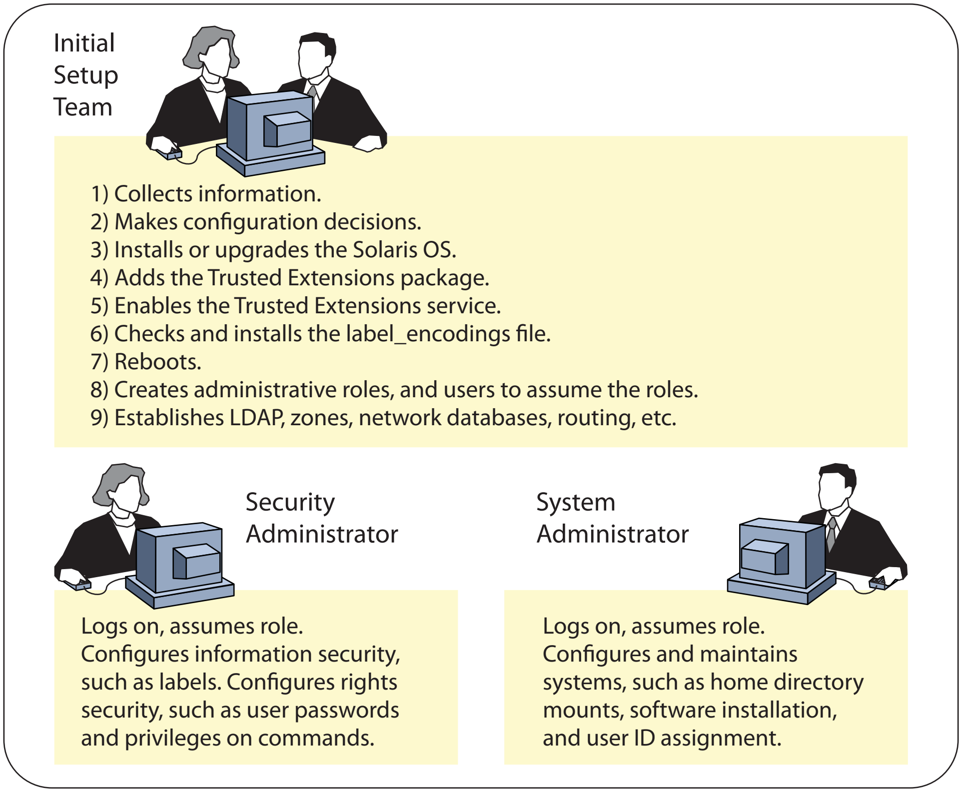


FIGURE 1–1 Administering a Trusted Extensions System: Task Division by Role

Collecting Information Before Enabling Trusted Extensions

As when configuring the Solaris OS, collect system, user, network, and label information before configuring Trusted Extensions. For details, see “Collect System Information Before Enabling Trusted Extensions” on page 48.

Backing Up the System Before Enabling Trusted Extensions

If your system has files that must be saved, perform a backup before enabling the Trusted Extensions software. The safest way to back up files is to do a level 0 dump. If you do not have a backup procedure in place, see the administrator's guide to your current operating system for instructions.

Note – If you are migrating from a Trusted Solaris 8 release, you can restore your data only if the Trusted Extensions labels are identical to the Trusted Solaris 8 labels. Because Trusted Extensions does not create multilevel directories, each file and directory on backup media is restored to a zone whose label is identical to the file label in the backup. Backup *must be completed* before you reboot the system with Trusted Extensions enabled.

Results of Enabling Trusted Extensions From an Administrator's Perspective

After the Trusted Extensions software is enabled and the system is rebooted, the following security features are in place. Many features are configurable by the security administrator.

- Auditing is enabled.
- A Sun label_encodings file is installed and configured.
- A trusted desktop, Solaris Trusted Extensions (GNOME), creates a windowing environment that enables Trusted Path workspaces in the global zone.
- As in the Solaris OS, rights profiles for roles are defined. As in the Solaris OS, roles are not defined.

  To use roles to administer Trusted Extensions, you must create the roles. During configuration, you create the Security Administrator role.
- Three Trusted Extensions network databases, `tnrhdb`, `tnrhtp`, and `tnzonecfg` are added. The databases are administered by using the Security Templates tool and the Trusted Network Zones tool in the Solaris Management Console.
- Trusted Extensions provides GUIs to administer the system. Some GUIs are extensions to a Solaris OS GUI.
  - A trusted editor enables administrators to modify local administrative files.
  - The Device Manager manages attached devices.
  - The Solaris Management Console provides Java-based tools to manage local and network administrative databases. The use of these tools is required for managing the trusted network, zones, and users.

CHAPTER 2

Configuration Roadmap for Trusted Extensions

This chapter outlines the tasks for enabling and configuring Solaris™ Trusted Extensions software.

Task Map: Preparing a Solaris System for Trusted Extensions

Ensure that the Solaris OS on which you plan to run Trusted Extensions supports the features of Trusted Extensions that you plan to use. Complete one of the two tasks that are described in the following task map.

Task	For Instructions
Prepare an existing or upgraded Solaris installation for Trusted Extensions.	"Prepare an Installed Solaris System for Trusted Extensions" on page 46
Install the Solaris OS with Trusted Extensions features in mind.	"Install a Solaris System to Support Trusted Extensions" on page 46

Task Map: Preparing For and Enabling Trusted Extensions

To prepare a Trusted Extensions system before configuring it, complete the tasks that are described in the following task map.

Task	For Instructions
Complete the preparation of your Solaris system.	"Task Map: Preparing a Solaris System for Trusted Extensions" on page 39

Task	For Instructions
Back up your system.	For a Trusted Solaris 8 system, back up the system as described in the documentation for your release. A labeled backup can be restored to each identically labeled zone.
	For a Solaris system, see *System Administration Guide: Basic Administration*.
Gather information and make decisions about your system and your Trusted Extensions network.	"Collecting Information and Making Decisions Before Enabling Trusted Extensions" on page 47
Enable Trusted Extensions.	"Enable Solaris Trusted Extensions" on page 50
Configure the system.	For a system with a monitor, see "Task Map: Configuring Trusted Extensions" on page 40.
	For a headless system, see "Headless System Configuration in Trusted Extensions (Task Map)" on page 119.
	For a Sun Ray™, see *Sun Ray Server Software 4.1 Installation and Configuration Guide for the Solaris Operating System*. For the Sun Ray 5 release, see the Sun Ray Server 4.2 and Sun Ray Connector 2.2 Documentation (`http://wikis.sun.com/display/SRS/Home`) web site. Together, this server and client comprise the *Sun Ray 5* package. To configure initial client-server communication, see "Configuring Trusted Network Databases (Task Map)" on page 270.
	For a laptop, go to the OpenSolaris Community: Security web page (`http://hub.opensolaris.org/bin/view/Community+Group+security/`). Click Trusted Extensions. On the Trusted Extensions page under Laptop Configurations, click `Laptop instructions`.
	To prevent networks from communicating with the global zone, configure the `vni0` interface. For an example, see the `Laptop instructions`. In the Solaris OS, you do not need to configure the `vni0` interface. By default, the `lo0` interface is an `all-zones` interface. You can make your external connection, `dhcp`, be an `all-zones` interface. For instructions, see the `Laptop instructions`.

Task Map: Configuring Trusted Extensions

For a secure configuration process, create roles early. The order of tasks when roles configure the system is shown in the following task map.

1. Configure the global zone.

Tasks	For Instructions
Protect machine hardware by requiring a password to change hardware settings.	"Controlling Access to System Hardware" in *System Administration Guide: Security Services*
Configure labels. Labels *must* be configured for your site. If you plan to use the default `label_encodings` file, you can skip this task.	"Check and Install Your Label Encodings File" on page 54
If you are running an IPv6 network, you modify the `/etc/system` file to enable IP to recognize labeled packets.	"Enable IPv6 Networking in Trusted Extensions" on page 56
If the CIPSO Domain of Interpretation (DOI) of your network nodes is different from `1`, specify the DOI in the `/etc/system` file.	"Configure the Domain of Interpretation" on page 57
Boot to activate a labeled environment. Upon login, you are in the global zone. The system's `label_encodings` file enforces mandatory access control (MAC).	"Reboot and Log In to Trusted Extensions" on page 58
Initialize the Solaris Management Console. This GUI is used to label zones, among other tasks.	"Initialize the Solaris Management Console Server in Trusted Extensions" on page 60
Create the Security Administrator role and other roles that you plan to use locally. You create these roles just as you would create them in the Solaris OS. You can delay this task until the end. For the consequences, see "Devising a Configuration Strategy for Trusted Extensions" on page 34.	"Creating Roles and Users in Trusted Extensions" on page 79 "Verify That the Trusted Extensions Roles Work" on page 88

Skip the next set of tasks if you are using local files administer the system.

2. Configure a naming service.

Tasks	For Instructions
If you plan to use files to administer Trusted Extensions, you can skip the following tasks.	No configuration is required for the files naming service.
If you have an existing Sun Java™ System Directory Server (LDAP server), add Trusted Extensions databases to the server. Then make your first Trusted Extensions system a proxy of the LDAP server. If you do not have an LDAP server, then configure your first system as the server.	Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)"
Manually set up an LDAP toolbox for the Solaris Management Console. The toolbox can be used to modify Trusted Extensions attributes on network objects.	"Configuring the Solaris Management Console for LDAP (Task Map)" on page 113
For systems that are not the LDAP server or proxy server, make them an LDAP client.	"Make the Global Zone an LDAP Client in Trusted Extensions" on page 63
In the LDAP scope, create the Security Administrator role and other roles that you plan to use. You can delay this task until the end. For the consequences, see "Devising a Configuration Strategy for Trusted Extensions" on page 34.	"Creating Roles and Users in Trusted Extensions" on page 79 "Verify That the Trusted Extensions Roles Work" on page 88

3. Create labeled zones.

Tasks	For Instructions
Run the `txzonemgr` command. Follow the menus to configure the network interfaces, then create and customize the first labeled zone. Then, clone the rest of the zones.	"Creating Labeled Zones" on page 65
(Optional) After all zones are successfully customized, add zone-specific network addresses and default routing to the labeled zones.	"Adding Network Interfaces and Routing to Labeled Zones" on page 71

The following tasks might be necessary in your environment.

4. Complete system setup.		
	Tasks	**For Instructions**
	Identify additional remote hosts that require a label, one or more multilevel ports, or a different control message policy.	"Configuring Trusted Network Databases (Task Map)" on page 270
	Create a multilevel home directory server, then automount the installed zones.	"Creating Home Directories in Trusted Extensions" on page 90
	Configure auditing, mount file systems, and perform other tasks before enabling users to log in to the system.	Part II, "Administration of Trusted Extensions"
	Add users from an NIS environment to your LDAP server.	"Add an NIS User to the LDAP Server" on page 93
	Add a host and its labeled zones to the LDAP server.	"Configuring Trusted Network Databases (Task Map)" on page 270

CHAPTER 3

Adding Solaris Trusted Extensions Software to the Solaris OS (Tasks)

This chapter describes how to prepare the Solaris OS for Solaris Trusted Extensions software. This chapter also describes the information you need before enabling Trusted Extensions. Instructions on how to enable Trusted Extensions is also provided.

Initial Setup Team Responsibilities

Trusted Extensions software is designed to be enabled and configured by two people with distinct responsibilities. However, the Solaris installation program does not enforce this two-role task division. Instead, task division is enforced by roles. Because roles and users are not created until after installation, it is a good practice to have an initial setup team of at least two people present to enable and configure Trusted Extensions software.

Installing or Upgrading the Solaris OS for Trusted Extensions

The choice of Solaris installation options can affect the use and security of Trusted Extensions:

- To properly support Trusted Extensions, you must install the underlying Solaris OS securely. For Solaris installation choices that affect Trusted Extensions, see "Install a Solaris System to Support Trusted Extensions" on page 46.
- If you have been using the Solaris OS, check your current configuration against the requirements for Trusted Extensions. For configuration choices that affect Trusted Extensions, see "Prepare an Installed Solaris System for Trusted Extensions" on page 46.

▼ Install a Solaris System to Support Trusted Extensions

This task applies to fresh installations of the Solaris OS. If you are upgrading, see "Prepare an Installed Solaris System for Trusted Extensions" on page 46.

Install the Trusted Extensions Package After upgrading to the latest Dev release, and rebooting the system, open the Package Manager again to get the Trusted Extensions package. Enter trusted in the Search text area to get a list of Trusted Extensions packages. Select trusted-extensions . Then select Install/Update. There is also a new trusted-nonglobal package which enumerates the initial set of packages required in a labeled brand zone to run the Trusted Desktop. This will be retrieved from the repository when you install your first zone.

1. **When installing the Solaris OS, create a user account and the `root` role account.**

   In Trusted Extensions, you use the `root` role to configure the system.

2. **Assign a different password to each account.**

3. **After the default installation of OpenSolaris 2010.03 is completed, start the Package Manager.**

4. **Install the Trusted Extensions package.**

   a. **For list of Trusted Extensions packages, type `trusted` in the Search text area.**

   b. **Select `trusted-extensions`.**

   c. **Select `Install/Update`.**

   The correct packages are installed on your system.

▼ Prepare an Installed Solaris System for Trusted Extensions

This task applies to Solaris systems that have been in use, and on which you plan to run Trusted Extensions. Also, to run Trusted Extensions on an upgraded Solaris system, follow this procedure. Other tasks that might modify an installed Solaris system can be done during Trusted Extensions configuration.

Before You Begin Trusted Extensions cannot be enabled in some Solaris environments:

- If your system is part of a cluster, Trusted Extensions cannot be enabled on the system.
- The enabling of Trusted Extensions in an alternate boot environment (BE) is not supported. Trusted Extensions can only be enabled in the current boot environment.

1 **If non-global zones are installed on your system, remove them.**

Trusted Extensions use branded zones.

2 **If your system does not have a `root` password, create one.**

Administration tools in Trusted Extensions require passwords. If the `root` user does not have a password, then `root` cannot configure the system.

Use the default `crypt_unix` password encryption method for the `root` user. For details, see "Managing Password Information" in *System Administration Guide: Security Services*.

Note – Users must not disclose their passwords to another person, as that person might then have access to the data of the user and will not be uniquely identified or accountable. Note that disclosure can be direct, through the user deliberately disclosing her/his password to another person, or indirect, for example, through writing it down, or choosing an insecure password. The Solaris OS provides protection against insecure passwords, but cannot prevent a user from disclosing her or his password, or from writing it down.

---

3 **If you have created an `xorg.conf` file, you need to modify it.**

Add the following line to the end of the Module section in the `/etc/X11/xorg.conf` file.

```
load "xtsol"
```

Note – By default, the `xorg.conf` file does not exist. Do nothing if this file does not exist.

---

4 **(Optional) Dedicate a partition for audit files.**

Trusted Extensions enables auditing by default. For audit files, best practice is to create a dedicated partition.

Collecting Information and Making Decisions Before Enabling Trusted Extensions

For each system on which Solaris Trusted Extensions is going to be configured, you need to know some information, and make some decisions about configuration. For example, because you are going to create labeled zones, you might want to set aside disk space where the zones can be cloned as a Solaris ZFS™ File System. Solaris ZFS provides additional isolation for the zones.

▼ Collect System Information Before Enabling Trusted Extensions

1 **Determine the system's main hostname and IP address.**

If you are using DHCP, skip this step.

The hostname is the name of the host on the network, and is the global zone. On a Solaris system, the `getent` command returns the hostname, as in:

```
# getent hosts machine1
192.168.0.11   machine1
```

2 **Determine the IP address assignments for labeled zones.**

If you are using DHCP, skip this step.

A system with two IP addresses can function as a multilevel server. A system with one IP address must have access to a multilevel server in order to print or perform multilevel tasks. For a discussion of IP address options, see "Planning for Multilevel Access" on page 32.

Most systems require a second IP address for the labeled zones. For example, the following is a host with a second IP address for labeled zones:

```
# getent hosts machine1-zones
192.168.0.12   machine1-zones
```

3 **Collect LDAP configuration information.**

For the LDAP server that is running Trusted Extensions software, you need the following information:

- The name of the Trusted Extensions domain that the LDAP server serves
- The IP address of the LDAP server
- The LDAP profile name that will be loaded

For an LDAP proxy server, you also need the password for the LDAP proxy.

▼ Make System and Security Decisions Before Enabling Trusted Extensions

For each system on which Solaris Trusted Extensions is going to be configured, make these configuration decisions before enabling the software.

1 **Decide how securely the system hardware needs to be protected.**

At a secure site, this step has been done for every installed Solaris system.

- For SPARC systems, a PROM security level and password has been provided.
- For x86 systems, the BIOS is protected.
- On all systems, `root` is protected with a password.

2 Prepare your `label_encodings` file.

If you have a site-specific `label_encodings` file, the file must be checked and installed before other configuration tasks can be started. If your site does not have a `label_encodings` file, you can use the default file that Sun supplies. Sun also supplies other `label_encodings` files, which you can find in the `/etc/security/tsol` directory. The Sun files are demonstration files. They might not be suitable for production systems.

To customize a file for your site, see *Solaris Trusted Extensions Label Administration*.

3 From the list of labels in your `label_encodings` file, make a list of the labeled zones that you need to create.

For the default `label_encodings` file, the labels are the following, and the zone names can be similar to the following:

Label	Zone Name
`PUBLIC`	`public`
`CONFIDENTIAL : INTERNAL`	`internal`
`CONFIDENTIAL : NEED TO KNOW`	`needtoknow`
`CONFIDENTIAL : RESTRICTED`	`restricted`

For ease of NFS mounting, the zone name of a particular label must be identical on every system. Some systems, such as multilevel print servers, do not need to have labeled zones installed. However, if you do install labeled zones on a print server, the zone names must be identical to the zone names of other systems on your network.

4 Decide when to create roles.

Your site's security policy can require you to administer Trusted Extensions by assuming a role. If so, or if you are configuring the system to satisfy criteria for an evaluated configuration, you must create roles early in the configuration process.

If you are not required to configure the system by using roles, you can choose to configure the system as superuser. This method of configuration is less secure. Audit records do not indicate which user was superuser during configuration. Superuser can perform all tasks on the system, while a role can perform a more limited set of tasks. Therefore, configuration is more controlled when being performed by roles.

5 Choose a zone creation method.

You can create zones from scratch or clone zones. These methods differ in speed of creation. For the trade-offs, see "Planning for Zones in Trusted Extensions" on page 31.

6 Plan your LDAP configuration.

Using local files for administration is practical for non-networked systems.

LDAP is the naming service for a networked environment. A populated LDAP server is required when you configure several machines.

- If you have an existing Sun Java™ System Directory Server (LDAP server), you can create an LDAP proxy server on a system that is running Trusted Extensions. The multilevel proxy server handles communications with the unlabeled LDAP server.
- If you do not have an LDAP server, you can configure a system that runs Trusted Extensions software as a multilevel LDAP server.

7 Decide other security issues for each system and for the network.

For example, you might want to consider the following security issues:

- Determine which devices can be attached to the system and allocated for use.
- Identify which printers at what labels are accessible from the system.
- Identify any systems that have a limited label range, such as a gateway system or a public kiosk.
- Identify which labeled systems can communicate with particular unlabeled systems.

Enabling the Solaris Trusted Extensions Service

In the Solaris OS, Solaris Trusted Extensions is a service that is managed by the service management facility (SMF). The name of the service is `svc:/system/labeld:default`. By default, the `labeld` service is disabled.

▼ Enable Solaris Trusted Extensions

The `labeld` service attaches labels to communications endpoints. For example, the following are labeled:

- All zones and the directories and files within each zone
- All processes including window processes
- All network communications

Before You Begin You have completed the tasks in "Installing or Upgrading the Solaris OS for Trusted Extensions" on page 45 and "Collecting Information and Making Decisions Before Enabling Trusted Extensions" on page 47.

1. **Open a terminal window and enable the `labeld` service.**

   ```
   # svcadm enable -s labeld
   ```

   The `labeld` service adds labels to the system and starts the Solaris auditing service and device allocation. Do not perform other tasks until the cursor returns to the prompt.

2. **Verify that the service is enabled.**

   ```
   # svcs -x labeld
   svc:/system/labeld:default (Trusted Extensions)
    State: online since weekday month date hour:minute:second year
      See: labeld(1M)
   Impact: None.
   ```

3. **If you plan to perform any of the following tasks, do not reboot:**

   - **Protect the hardware.**
   - **Install your own `label_encodings` file.**
   - **Run on an IPv6 network.**
   - **Modify the CIPSO DOI.**

     To perform any of these tasks, see "Setting Up the Global Zone in Trusted Extensions" on page 53. You will reboot after these tasks are accomplished.

4. **Otherwise, reboot.**

CHAPTER 4

Configuring Trusted Extensions (Tasks)

This chapter covers how to configure Solaris™ Trusted Extensions on a system with a monitor. To work properly, Trusted Extensions software requires configuration of the following: labels, zones, the network, users who can assume roles, roles, and tools.

- "Setting Up the Global Zone in Trusted Extensions" on page 53
- "Creating Labeled Zones" on page 65
- (Optional) "Adding Network Interfaces and Routing to Labeled Zones" on page 71
- "Creating Roles and Users in Trusted Extensions" on page 79
- "Creating Home Directories in Trusted Extensions" on page 90
- "Adding Users and Hosts to an Existing Trusted Network" on page 93
- "Troubleshooting Your Trusted Extensions Configuration" on page 95
- "Additional Trusted Extensions Configuration Tasks" on page 98

For other configuration tasks, see Part II, "Administration of Trusted Extensions."

Setting Up the Global Zone in Trusted Extensions

Before setting up the global zone, you must make decisions about your configuration. For the decisions, see "Collecting Information and Making Decisions Before Enabling Trusted Extensions" on page 47.

Task	Description	For Instructions
Protect the hardware.	Hardware can be protected by requiring a password to change hardware settings.	"Controlling Access to System Hardware" in *System Administration Guide: Security Services*
Configure labels.	Labels *must* be configured for your site. If you plan to use the default `label_encodings` file, you can skip this step.	"Check and Install Your Label Encodings File" on page 54

Task	Description	For Instructions
For IPv6, modify the `/etc/system` file.	If you are running an IPv6 network, you modify the `/etc/system` file to enable IP to recognize labeled packets.	"Enable IPv6 Networking in Trusted Extensions" on page 56
For a DOI whose value is not `1`, modify the `/etc/system` file.	If the CIPSO Domain of Interpretation (DOI) of your network nodes is different from `1`, specify the DOI in the `/etc/system` file.	"Configure the Domain of Interpretation" on page 57
Reboot and log in.	Upon login, you are in the global zone, which is an environment that recognizes and enforces mandatory access control (MAC).	"Reboot and Log In to Trusted Extensions" on page 58
Initialize the Solaris Management Console.	Trusted Extensions adds tools to the Solaris Management Console for administering users, roles, zones, and the network.	"Initialize the Solaris Management Console Server in Trusted Extensions" on page 60
Configure LDAP.	If you are using the LDAP naming service, set up the LDAP service.	Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)"
	If you have set up the LDAP service, make this system an LDAP client.	"Make the Global Zone an LDAP Client in Trusted Extensions" on page 63

▼ Check and Install Your Label Encodings File

Your encodings file must be compatible with any Trusted Extensions host with which you are communicating.

Note – Trusted Extensions installs a default `label_encodings` file. This default file is useful for demonstrations. However, this file might not be a good choice for your use. If you plan to use the default file, you can skip this procedure.

- If you are familiar with encodings files, you can use the following procedure.
- If you are not familiar with encodings files, consult *Solaris Trusted Extensions Label Administration* for requirements, procedures, and examples.

Caution – You *must* successfully install labels before continuing, or the configuration will fail.

Before You Begin You are the security administrator. The security administrator is responsible for editing, checking, and maintaining the `label_encodings` file. If you plan to edit the `label_encodings` file, make sure that the file itself is writable. For more information, see the `label_encodings(4)` man page.

1 **Insert the media with the `label_encodings` file into the appropriate device.**

2 Copy the `label_encodings` file to the disk.

3 Check the syntax of the file and make it the active `label_encodings` file.

Use the command line.

a. Open a terminal window.

b. Run the `chk_encodings` command.

```
# /usr/sbin/chk_encodings /full-pathname-of-label-encodings-file
```

c. Read the output and do one of the following:

- **Resolve errors.**

  If the command reports errors, the errors *must* be resolved before continuing. For assistance, see Chapter 3, "Creating a Label Encodings File (Tasks)," in *Solaris Trusted Extensions Label Administration*

- **Make the file the active `label_encodings` file.**

```
# cp /full-pathname-of-label-encodings-file \
 /etc/security/tsol/label.encodings.site
# cd /etc/security/tsol
# cp label_encodings label_encodings.tx.orig
# cp label.encodings.site label_encodings
```

Caution – Your `label_encodings` file *must* pass the Check Encodings test before you continue.

Example 4–1 Checking `label_encodings` Syntax on the Command Line

In this example, the administrator tests several `label_encodings` files by using the command line.

```
# /usr/sbin/chk_encodings /var/encodings/label_encodings1
No errors found in /var/encodings/label_encodings1
# /usr/sbin/chk_encodings /var/encodings/label_encodings2
No errors found in /var/encodings/label_encodings2
```

When management decides to use the `label_encodings2` file, the administrator runs a semantic analysis of the file.

```
# /usr/sbin/chk_encodings -a /var/encodings/label_encodings2
No errors found in /var/encodings/label_encodings2

---> VERSION = MYCOMPANY LABEL ENCODINGS  2.0 10/10/2006
```

```
---> CLASSIFICATIONS <---

   Classification 1: PUBLIC
   Initial Compartment bits: 10
   Initial Markings bits: NONE

---> COMPARTMENTS AND MARKINGS USAGE ANALYSIS <---
...
---> SENSITIVITY LABEL to COLOR MAPPING <---
...
```

The administrator prints a copy of the semantic analysis for her records, then moves the file to the `/etc/security/tsol` directory.

```
# cp /var/encodings/label_encodings2 /etc/security/tsol/label.encodings.10.10.06
# cd /etc/security/tsol
# cp label_encodings label_encodings.tx.orig
# cp label.encodings.10.10.06 label_encodings
```

Finally, the administrator verifies that the `label_encodings` file is the company file.

```
# /usr/sbin/chk_encodings -a /etc/security/tsol/label_encodings | head -4
No errors found in /etc/security/tsol/label_encodings

---> VERSION = MYCOMPANY LABEL ENCODINGS  2.0 10/10/2006
```

▼ Enable IPv6 Networking in Trusted Extensions

CIPSO options do not have an Internet Assigned Numbers Authority (IANA) number to use in the IPv6 Option Type field of a packet. The entry that you set in this procedure supplies a number to use on the local network until IANA assigns a number for this option. Trusted Extensions disables IPv6 networking if this number is not defined.

To enable an IPv6 network in Trusted Extensions, you must add an entry in the `/etc/system` file.

- **Type the following entry into the `/etc/system` file:**

  ```
  set ip:ip6opt_ls = 0x0a
  ```

Troubleshooting

- If error messages during boot indicate that your IPv6 configuration is incorrect, correct the entry:
  - Check that the entry is spelled correctly.
  - Check that the system has been rebooted after adding the correct entry to the `/etc/system` file.

- If you install Trusted Extensions on a Solaris system that currently has IPv6 enabled, but you fail to add the IP entry in `/etc/system`, you see the following error message: `t_optmgmt: System error: Cannot assign requested address` *time-stamp*
- If you install Trusted Extensions on a Solaris system that does not have IPv6 enabled, and you fail to add the IP entry in `/etc/system`, you see the following types of error messages:
  - `WARNING: IPv6 not enabled via /etc/system`
  - `Failed to configure IPv6 interface(s): hme0`
  - `rpcbind: Unable to join IPv6 multicast group for rpc broadcast` *broadcast-number*

▼ Configure the Domain of Interpretation

All communications to and from a system that is configured with Trusted Extensions must follow the labeling rules of a single CIPSO Domain of Interpretation (DOI). The DOI that is used in each message is identified by an integer number in the CIPSO IP Option header. By default, the DOI in Trusted Extensions is `1`.

If your DOI is not `1`, you must add an entry to the `/etc/system` file and modify the `doi` value in the default security templates.

1 **Type your DOI entry into the `/etc/system` file:**

```
set default_doi = n
```

This positive, non-zero number must match the DOI number in the `tnrhtp` database for your node and for the systems that your node communicates with.

2 **Before adding the `tnrhtp` database to your LDAP server, modify the `doi` value in the default entries and all entries for local addresses.**

Trusted Extensions provides two templates in the `tnrhtp` database, `cipso` and `admin_low`. If you have added entries for local addresses, also modify these entries.

a. **Open the `tnrhtp` database in the trusted editor.**

```
# /usr/dt/bin/trusted_edit /etc/security/tsol/tnrhtp
```

b. **Copy the `cipso` template entry to another line.**

```
cipso:host_type=cipso;doi=1;min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH
cipso:host_type=cipso;doi=1;min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH
```

c. **Comment out one of the `cipso` entries.**

```
#cipso:host_type=cipso;doi=1;min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH
cipso:host_type=cipso;doi=1;min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH
```

d. **Modify the `doi` value in the uncommented `cipso` entry.**

Make this value the same as the `default_doi` value in the `/etc/system` file.

```
#cipso:host_type=cipso;doi=1;min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH
cipso:host_type=cipso;doi=n;min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH
```

e. **Change the `doi` value for the `admin_low` entry.**

```
#admin_low:host_type=unlabeled;min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH;doi=1;def_label=ADMIN_LOW
admin_low:host_type=unlabeled;min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH;doi=n;def_label=ADMIN_LOW
```

You are finished when every `doi` value in every entry in the `tnrhtp` database is the same.

Troubleshooting If the `/etc/system` file sets a `default_doi` value other than `1`, and a security template for this system sets a value that does not match this `default_doi` value, then messages similar to the following are displayed on the system console during interface configuration:

- `NOTICE: er10 failed: 10.17.1.12 has wrong DOI 4 instead of 1`
- `Failed to configure IPv4 interface(s): er10`

Interface configuration failure can result in login failure:

- `Hostname: unknown`
- `unknown console login: root`
- `Oct 10 10:10:20 unknown login: pam_unix_cred: cannot load hostname Error 0`

To correct the problem, boot the system into single-user mode and correct the security templates as described in this procedure.

See Also For more information about the DOI, see "Network Security Attributes in Trusted Extensions" on page 254.

To change the `doi` value in the security templates that you create, see "How to Construct a Remote Host Template" on page 272.

To use the editor of your choice as the trusted editor, see "How to Assign the Editor of Your Choice as the Trusted Editor" on page 166.

▼ Reboot and Log In to Trusted Extensions

At most sites, two or more administrators, who serve as an initial setup team, are present when configuring the system.

Before You Begin Before you first log in, become familiar with the desktop and label options in Trusted Extensions. For details, see Chapter 3, "Logging In to Trusted Extensions (Tasks)," in *Solaris Trusted Extensions User's Guide*.

1 **Reboot the system.**

```
# /usr/sbin/reboot
```

If your system does not have a graphical display, go to Chapter 6, "Configuring a Headless System With Trusted Extensions (Tasks)."

2 **Log in to the Solaris Trusted Extensions (GNOME) desktop as the user account that you created during installation.**

a. **In the login window, select the labeled desktop.**

i. **Select Options > Select Session ... Solaris Trusted Extensions (GNOME).**

ii. **Then select Change Session > Make Default Login.**

b. **In the login dialog box, type *username* and the user's password.**

Users must not disclose their passwords to another person, as that person might then have access to the data of the user and will not be uniquely identified or accountable. Note that disclosure can be direct, through the user deliberately disclosing his/her password to another person, or indirect, such as through writing it down, or choosing an insecure password. Trusted Extensions software provides protection against insecure passwords, but cannot prevent a user disclosing his/her password or writing it down.

3 **Use the mouse to dismiss the Status window and the Clearance window.**

GNOME will complain four times, once per workspace, that the label `PUBLIC` has no matching zone. Dismiss each of the four dialogs.

4 **Use the mouse to dismiss four complaint dialog boxes.**

GNOME complains four times, one complaint for each workspace, that the label `PUBLIC` has no matching zone.

5 **Assume the root role.**

a. **Switch to fourth workspace.**

b. **Click the trusted stripe where your name is displayed.**

The `root` role appears in a pulldown menu.

c. **Click the `root` role.**

Note – You must log off or lock the screen before leaving a system unattended. Otherwise, a person can access the system without having to pass identification and authentication, and that person would not be uniquely identified or accountable.

▼ Initialize the Solaris Management Console Server in Trusted Extensions

This procedure enables you to administer users, roles, hosts, zones, and the network on this system. On the first system that you configure, only the `files` scope is available.

Before You Begin You must be superuser.

To use the LDAP toolbox on the LDAP server from a Solaris Management Console that is running on a client, you must complete all of the tasks in "Configuring the Solaris Management Console for LDAP (Task Map)" on page 113.

1. **Start the Solaris Management Console.**

```
# /usr/sbin/smc &
```

Note – The first time the Solaris Management Console is started, it performs several registration tasks. These tasks can take a few minutes.

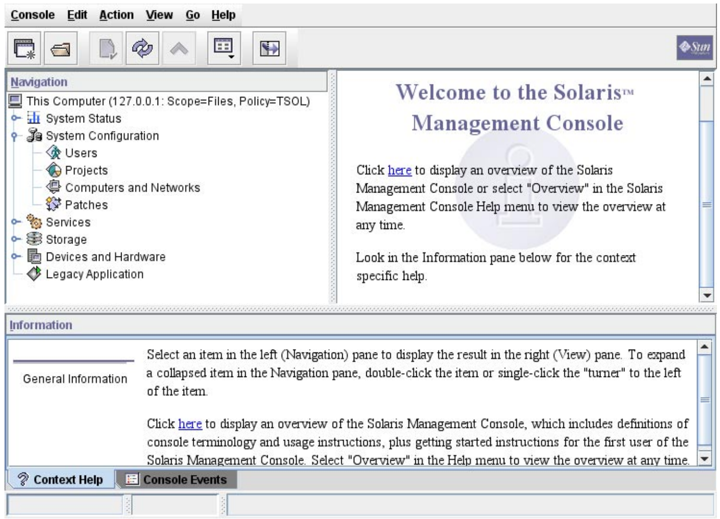


FIGURE 4–1 Solaris Management Console Initial Window

2 Do one of the following if toolbox icons do not appear in the Solaris Management Console:

- **If the Navigation pane is not visible:**

  a. In the Open Toolbox dialog box that is displayed, click Load next to this system's name under Server.

  If this system does not have the recommended amount of memory and swap, it might take a few minutes for the toolboxes to display. For recommendations, see "Installing or Upgrading the Solaris OS for Trusted Extensions" on page 45.

  b. From the list of toolboxes, select a toolbox whose `Policy=TSOL`.

  Figure 4–2 shows a This Computer (*this-host*: `Scope=Files, Policy=TSOL`) toolbox. Trusted Extensions modifies tools under the System Configuration node.

  Caution – Do not choose a toolbox that has no policy. Toolboxes without a listed policy do not support Trusted Extensions.

  Your toolbox choice depends on which scope you want to influence.

  - To edit local files, choose the Files scope.
  - To edit LDAP databases, choose the LDAP scope.

    After you complete all of the tasks in "Configuring the Solaris Management Console for LDAP (Task Map)" on page 113, the LDAP scope is available.

  c. Click Open.

- **If the Navigation pane is visible, but the toolbox icons are stop signs:**

  a. **Exit the Solaris Management Console.**

  b. **Restart the Solaris Management Console.**

```
# /usr/sbin/smc &
```

3 **If you have not yet done so, select a toolbox whose `Policy=TSOL`.**

The following figure shows a This Computer (*this-host*: `Scope=Files, Policy=TSOL`) toolbox. Trusted Extensions modifies tools under the System Configuration node.

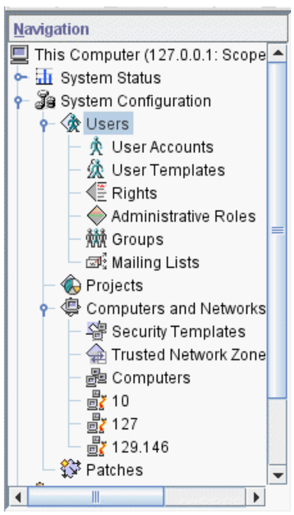


FIGURE 4–2 Trusted Extensions Tools in the Solaris Management Console

4 **(Optional) Save the current toolbox.**

Saving a `Policy=TSOL` toolbox enables a Trusted Extensions toolbox to load by default. Preferences are saved per role, per host. The host is the Solaris Management Console server.

a. **From the Console menu, choose Preferences.**

The Home toolbox is selected.

b. **Define a `Policy=TSOL` toolbox as the Home toolbox.**

Put the current toolbox in the Location field by clicking the Use Current Toolbox button.

c. **Click OK to save the preferences.**

5 **Exit the Solaris Management Console.**

See Also For an overview of the Trusted Extensions additions to the Solaris Management Console, see "Solaris Management Console Tools" on page 141. To use the Solaris Management Console to create security templates, see "Configuring Trusted Network Databases (Task Map)" on page 270.

Make the Global Zone an LDAP Client in Trusted Extensions

For LDAP, this procedure establishes the naming service configuration for the global zone. If you are not using LDAP, you can skip this procedure.

Use the `txzonemgr` script.

Note – If you plan to set up a name server in each labeled zone, you are responsible for establishing the LDAP client connection to each labeled zone.

Before You Begin The Sun Java™ System Directory Server, that is, the LDAP server, must exist. The server must be populated with Trusted Extensions databases, and this system must be able to contact the server. So, the system that you are configuring must have an entry in the `tnrhdb` database on the LDAP server, or this system must be included in a wildcard entry before you perform this procedure.

If an LDAP server that is configured with Trusted Extensions does not exist, you must complete the procedures in Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)," before you perform this procedure.

1. **If you are using DNS, modify the `nsswitch.ldap` file.**

   a. **Save a copy of the original `nsswitch.ldap` file.**

   The standard naming service switch file for LDAP is too restrictive for Trusted Extensions.

```
# cd /etc
# cp nsswitch.ldap nsswitch.ldap.orig
```

   b. **Change the `nsswitch.ldap` file entries for the following services.**

   The correct entries are similar to the following:

```
hosts:    files dns ldap

ipnodes:    files dns ldap

networks:   ldap files
protocols:  ldap files
rpc:        ldap files
```

```
ethers:     ldap files
netmasks:   ldap files
bootparams: ldap files
publickey:  ldap files

services:   files
```

Note that Trusted Extensions adds two entries:

```
tnrhtp:    files ldap
tnrhdb:    files ldap
```

c. **Copy the modified `nsswitch.ldap` file to `nsswitch.conf`.**

```
# cp nsswitch.ldap nsswitch.conf
```

2 **To create an LDAP client, use the `txzonemgr` script.**

The Create LDAP Client menu item configures the global zone only.

a. **Follow the instructions in** **"Run the `txzonemgr` Script" on page 66****.**

The title of the dialog box is Labeled Zone Manager.

b. **Select Create LDAP Client.**

c. **Answer the following prompts and click OK after each answer:**

`Enter Domain Name:`	*Type the domain name*
`Enter Hostname of LDAP Server:`	*Type the name of the server*
`Enter IP Address of LDAP Server` *servername*`:`	*Type the IP address*
`Enter LDAP Proxy Password:`	*Type the password to the server*
`Confirm LDAP Proxy Password:`	*Retype the password to the server*
`Enter LDAP Profile Name:`	*Type the profile name*

d. **Confirm or cancel the displayed values.**

```
Proceed to create LDAP Client?
```

When you confirm, the `txzonemgr` script adds the LDAP client. Then, a window displays the command output.

3 **Verify that the information on the server is correct.**

a. **Open a terminal window, and query the LDAP server.**

```
# ldapclient list
```

The output looks similar to the following:

```
NS_LDAP_FILE_VERSION= 2.0
NS_LDAP_BINDDN= cn=proxyagent,ou=profile,dc=domain-name
```

```
...
NS_LDAP_BIND_TIME= number
```

b. Correct any errors.

If you get an error, create the LDAP client again and supply the correct values. For example, the following error can indicate that the system does not have an entry on the LDAP server:

```
LDAP ERROR (91): Can't connect to the LDAP server.
Failed to find defaultSearchBase for domain domain-name
```

To correct this error, you need to check the LDAP server.

Example 4–2 Using Host Names After Loading a `resolv.conf` File

In this example, the administrator wants a particular set of DNS servers to be available to the system. The administrator copies a `resolv.conf` file from a server on a trusted net. Because DNS is not yet active, the administrator uses the server's IP address to locate the server.

```
# cd /etc
# cp /net/10.1.1.2/export/txsetup/resolv.conf resolv.conf
```

After the `resolv.conf` file is copied and the `nsswitch.conf` file includes `dns` in the `hosts` entry, the administrator can use host names to locate systems.

Creating Labeled Zones

The instructions in this section configure labeled zones on a system that has been assigned at most two IP addresses. For other configurations, see the configuration options in "Task Map: Preparing For and Enabling Trusted Extensions" on page 39.

Task	Description	For Instructions
1. Run the `txzonemgr` script.	The `txzonemgr` script creates a GUI that presents the appropriate tasks as you configure your zones.	"Run the `txzonemgr` Script" on page 66
2. Create, install, boot, and halt the first zone.	In the default configuration, create the `PUBLIC` zone. This zone forms the template for other labeled zones.	"Create the First Labeled Zone" on page 66
3. Manage network interfaces in the global zone.	Configure interfaces in the global zone, or create logical interfaces and configure them in the global zone.	"Configure the Network Interfaces in Trusted Extensions" on page 67
4. Create a clone.	Clone the first zone. The clone is not assigned a label.	"Clone the First Zone in Trusted Extensions" on page 68
5. Verify that the first zone is working correctly.	Test connection with a non-Trusted Extensions system.	"Verify the Status of the Public Zone" on page 69

Task	Description	For Instructions
6. Label the cloned zone.	Add a label to a cloned zone.	"Adding Network Interfaces and Routing to Labeled Zones" on page 71
7. Create a zone from a snapshot.	Create the rest of the zones.	"Create a Zone From the Snapshot" on page 70
8. Create a labeled working environment.	Activate the `PUBLIC` and `INTERNAL` workspaces.	"Activate Two Zone Workspaces" on page 71

▼ Run the `txzonemgr` Script

This script steps you through the tasks to properly configure, install, initialize, and boot labeled zones. In the script, you name each zone, associate the name with a label, install the packages to create a virtual OS, and then boot the zone to start services in that zone. The script includes copy zone and clone zone tasks. You can also halt a zone, change the state of a zone, and add zone-specific network interfaces.

This script presents a dynamically-determined menu that displays only valid choices for the current circumstances. For instance, if the status of a zone is configured, the Install zone menu item is not displayed. Tasks that are completed do not display in the list.

Before You Begin You have assumed the `root` role.

1. **Open a terminal window in the fourth workspace.**

2. **Run the `txzonemgr` script.**

```
# /usr/sbin/txzonemgr
```

The script opens the `Labeled Zone Manager` dialog box. This `zenity` dialog box prompts you for the appropriate tasks, depending on the current state of your installation.

To perform a task, you select the menu item, then press the Return key or click OK. When you are prompted for text, type the text then press the Return key or click OK.

Tip – To view the current state of zone completion, click Return to Main Menu in the Labeled Zone Manager.

▼ Create the First Labeled Zone

You do not have to create a zone for every label in your `label_encodings` file, but you can. The administrative GUIs enumerate the labels that can have zones created for them on this system.

Before You Begin You are in the `root` role. The Labeled Zone Manager dialog box is displayed. To open this GUI, see "Run the `txzonemgr` Script" on page 66.

You have not created a zone yet.

1. **Click OK to the following dialog box:**

   ```
   Do you want to create the public zone using default settings?
   ```

   After the public zone is created, another terminal window appears. Its title is `Zone Terminal Console: public`. The public zone boots, initializes, and then prompts for the `root` password.

2. **Press the F2 key twice to provide the password for the `root` role.**

   The zone reboots.

   The Labeled Zone Manager dialog box displays the state and options for the public zone.

3. **Halt the public zone by selecting Halt from the Labeled Zone Manager.**

   In the Zone Terminal Console window, a notice appears: `Notice: Zone Halted`

4. **From the public zone options list, click `Select another zone...`**

▼ Configure the Network Interfaces in Trusted Extensions

In this task, you configure the networking in the global zone. You must create exactly one `all-zones` interface. An `all-zones` interface is shared by the labeled zones and the global zone. The shared interface is used to route traffic between the labeled zones and the global zone. To configure this interface, do one of the following:

- Create a logical interface from a physical interface, then share the physical interface.

  This configuration is the simplest to administer. Choose this configuration when your system has been assigned two IP addresses. In this procedure, the logical interface becomes the global zone's specific address, and the physical interface is shared between the global zone and the labeled zones.
- Share a physical interface

  Choose this configuration when your system has been assigned one IP address. In this configuration, the physical interface is shared between the global zone and the labeled zones.
- Share a virtual network interface, `vni0`

  Choose this configuration when you are configuring DHCP, or when each subnetwork is at a different label. For a sample procedure, refer to the laptop instructions in the Trusted Extensions section of OpenSolaris Community: Security web page (`http://hub.opensolaris.org/bin/view/Community+Group+security/`).

In the Solaris Express Community Edition, the loopback interface in Trusted Extensions is created as an all-zones interface. Therefore, you do not need to create a `vni0` shared interface.

To add zone-specific network interfaces, finish and verify zone creation before adding the interfaces. For the procedure, see "Add a Network Interface to Route an Existing Labeled Zone" on page 72.

Before You Begin The public zone is halted.

The Labeled Zone Manager is displayed. To open this GUI, see "Run the `txzonemgr` Script" on page 66.

From the public zone options list, you have clicked `Select another zone...`

1. **In the Labeled Zone Manager, select the global zone.**

2. **Select Configure Network Interfaces.**

   A list of interfaces is displayed. Look for an interface that is listed with the following characteristics:

   - Type of `physical`
   - IP address of your hostname
   - Template of `cipso`
   - State of `Up`

3. **Select the interface that corresponds to your hostname.**

4. **From the list of commands, select** `Share with Shared-IP Zones`**.**

5. **Click Cancel to return to the global zone command list**

6. **To connect to other systems on your network that are running Trusted Extensions, select Add Multilevel Access to Remote Host...**

   a. **Type the IP address of another Trusted Extensions system.**

   b. **Run the corresponding commands on the other Trusted Extensions system.**

▼ Clone the First Zone in Trusted Extensions

Before You Begin You have completed "Create the First Labeled Zone" on page 66 and "Configure the Network Interfaces in Trusted Extensions" on page 67. The public zone is still halted.

The Labeled Zone Manager dialog box is displayed. To open this GUI, see "Run the `txzonemgr` Script" on page 66.

1. **From the Labeled Zone Manager, select Create a new zone...**

   You are prompted to Enter Zone Name.

2. **Type** `snapshot` **as the zone name.**

   A list of options appear for the `snapshot` zone.

3. **Select Clone...**

   A list of installed zones appears. The list includes the name `public`.

4. **Double-click** `public`**.**

   The `snapshot` zone does not install automatically, so select Set Manual Booting The snapshot zone doesn't need a label since it is never booted. Verify the Boot option is not available.

5. **Select** `Set Manual Booting`**.**

   The `snapshot` zone is never booted, therefore it does not need a label. Verify that the `Boot` option is not available.

Verify the Status of the Public Zone

Note – The X server runs in the global zone. Each labeled zone must be able to connect with the global zone to use the X server. Therefore, zone networking must work before a zone can be used. For background information, see "Planning for Multilevel Access" on page 32.

Before You Begin The Labeled Zone Manager dialog box displays the global zone.

1. **Select** `Select another zone` **and choose** `public`**.**

   Enter the IP address of a system on your network not running TX. Then enter Boot You see the zone booting messages in the Zone Console window. Login as root, and run ifconfig -a Verify that the primary interface and IP address are available in this zone. Verify that you can ping the host to which you previously added remote access. Now logout and close the Zone Console window.

2. **Select** `Add Single-level Access to Remote Host...`

   a. **At the prompt, type the IP address of a system on your network that is not running Trusted Extensions.**

b. **Select Boot.**

Zone booting messages appear in the Zone Console Terminal window.

3 **In the public: Zone Console Terminal window, log in as `root`.**

4 **Run the `ifconfig -a` command.**

```
# ifconfig -a
```

Verify that the primary interface and IP address are available in this zone.

5 **Verify that you can `ping` the host to which you previously added single-level access.**

```
# ping remote-single-level-host
```

6 **Log out and close the Zone Console Terminal window.**

▼ Create a Zone From the Snapshot

This procedure creates the `internal` zone. Use this procedure to create the rest of your labeled zones.

Before You Begin You are in the `root` role. The Labeled Zone Manager dialog box is displayed. To open this GUI, see "Run the `txzonemgr` Script" on page 66.

You have completed "Clone the First Zone in Trusted Extensions" on page 68.

1 **In the Labeled Zone Manager, select `Select another zone`.**

2 **Choose global.**

3 **Select `Create a new zone:`**

The prompt, Enter Zone Name:, appears

4 **Type `internal`.**

A one-item list for the `internal` zone appears.

5 **Choose `Select Label...`.**

6 **From the label selection dialog box, select `INTERNAL USE ONLY` from the Sensitivity column and click OK.**

7 **In the list of options for the internal zone, select `Clone...`.**

8 **Select `snapshot` from the list of installed zones.**

`snapshot`is the only item in the list.

9 **Select `Boot`.**

▼ Activate Two Zone Workspaces

This procedure creates two labeled workspaces and opens a labeled window in a labeled workspace

Before You Begin You have completed “Create a Zone From the Snapshot” on page 70.

1 **Go to the first workspace.**

The desktop background should appear.

2 **Open a terminal window.**

The window is labeled `PUBLIC`.

3 **Create a workspace with a different label.**

a. **Switch to the second workspace.**

b. **Right-click and select Change Workspace Label...**

c. **Select INTERNAL USE ONLY and click OK.**

4 **Open a terminal window.**

The window is labeled `CONFIDENTIAL : INTERNAL USE ONLY`.

Adding Network Interfaces and Routing to Labeled Zones

The following tasks support environments where each zone is connected to a separate physical network.

Task	Description	For Instructions
EITHER 1a: Add a network interface to each labeled zone and use the global zone to reach the external network.	Connects each labeled zone to a separate physical network. The labeled zones use the network routing that the global zone provides.	“Add a Network Interface to Route an Existing Labeled Zone” on page 72

Task	Description	For Instructions
OR 1b: Add a network interface to each labeled zone with a default route.	Connects each zone to a separate physical network. The labeled zones do *not* use the global zone for routing.	"Add a Network Interface That Does Not Use the Global Zone to Route an Existing Labeled Zone" on page 74
2. Create a name service cache in each labeled zone.	Configures a name service daemon for each zone.	"Configure a Name Service Cache in Each Labeled Zone" on page 78

▼ Add a Network Interface to Route an Existing Labeled Zone

This procedure adds zone-specific network interfaces to existing labeled zones. This configuration supports environments where each labeled zone is connected to a separate physical network. The labeled zones use the network routing that the global zone provides.

Note – The global zone must configure an IP address for every subnet in which a non-global zone address is configured.

Before You Begin You are superuser in the global zone.

For every zone, you have completed the tasks in "Creating Labeled Zones" on page 65.

1. **In the global zone, type the IP addresses and hostnames for the additional network interfaces into the `/etc/hosts` file.**

   Use a standard naming convention, such as adding *-zone-name* to the name of the host.

   ```
   ## /etc/hosts in global zone
   10.10.8.2    hostname-zone-name1
   10.10.8.3    hostname-global-name1
   10.10.9.2    hostname-zone-name2
   10.10.9.3    hostname-global-name2
   ```

2. **For the network for each interface, add entries to the `/etc/netmasks` file.**

   ```
   ## /etc/netmasks in global zone
   10.10.8.0 255.255.255.0
   10.10.9.0 255.255.255.0
   ```

   For more information, see the `netmasks(4)` man page.

3. **In the global zone, plumb the zone-specific physical interfaces.**

   a. **Identify the physical interfaces that are already plumbed.**

   ```
   # ifconfig -a
   ```

b. **Configure the global zone addresses on each interface.**

```
# ifconfig interface-nameN1 plumb
# ifconfig interface-nameN1 10.10.8.3 up
# ifconfig interface-nameN2 plumb
# ifconfig interface-nameN2 10.10.9.3 up
```

c. **For each global zone address, create a `hostname.`*interface-nameN* file.**

```
# /etc/hostname.interface-nameN1
10.10.8.3
# /etc/hostname.interface-nameN2
10.10.9.3
```

The global zone addresses are configured immediately upon system startup. The zone-specific addresses are configured when the zone is booted.

4 **Assign a security template to each zone-specific network interface.**

If the gateway to the network is not configured with labels, assign the `admin_low` security template. If the gateway to the network is labeled, assign a `cipso` security template.

You can create security templates of host type `cipso` that reflect the label of every network. For the procedures to create and assign the templates, see "Configuring Trusted Network Databases (Task Map)" on page 270.

5 **Halt every labeled zone to which you plan to add a zone-specific interface.**

```
# zoneadm -z zone-name halt
```

6 **Start the Labeled Zone Manager.**

```
# /usr/sbin/txzonemgr
```

7 **For each zone where you want to add a zone-specific interface, do the following:**

a. **Select the zone.**

b. **Select Add Network.**

c. **Name the network interface.**

d. **Type the IP address of the interface.**

8 **In the Labeled Zone Manager for every completed zone, select Zone Console.**

9 **Select Boot.**

10 **In the Zone Console, verify that the interfaces have been created.**

```
# ifconfig -a
```

11 Verify that the zone has a route to the gateway for the subnet.

```
# netstat -rn
```

Troubleshooting To debug zone configuration, see the following:

- Chapter 26, "Troubleshooting Miscellaneous Solaris Zones Problems," in *System Administration Guide: Virtualization Using the Solaris Operating System*
- "Troubleshooting Your Trusted Extensions Configuration" on page 95
- "Troubleshooting the Trusted Network (Task Map)" on page 294

▼ Add a Network Interface That Does Not Use the Global Zone to Route an Existing Labeled Zone

This procedure sets zone-specific default routes for existing labeled zones. In this configuration, the labeled zones do *not* use the global zone for routing.

The labeled zone must be plumbed in the global zone before the zone is booted. However, to isolate the labeled zone from the global zone, the interface must be in the `down` state when the zone is booted. For more information, see Chapter 16, "Non-Global Zone Configuration (Overview)," in *System Administration Guide: Virtualization Using the Solaris Operating System*.

Note – A unique default route must be configured for every non-global zone that is booted.

Before You Begin You are superuser in the global zone.

For every zone, you have completed the tasks in "Creating Labeled Zones" on page 65. You are using either the `vni0` interface or the `lo0` interface to connect the labeled zones to the global zone.

1 For every network interface, determine its IP address, netmask, and default router.

Use the `ifconfig -a` command to determine the IP address and netmask. Use the `zonecfg -z zonename info net` command to determine if a default router has been assigned.

2 Create an empty `/etc/hostname.`*interface* file for each labeled zone.

```
# touch /etc/hostname.interface
# touch /etc/hostname.interface:n
```

For more information, see the `netmasks(4)` man page.

3 **Plumb the network interfaces of the labeled zones.**

```
# ifconfig zone1-network-interface plumb
# ifconfig zone2-network-interface plumb
```

4 **Verify that the labeled zone's interfaces are in the `down` state.**

```
# ifconfig -a
zone1-network-interface  zone1-IP-address down
zone2-network-interface  zone2-IP-address down
```

The zone-specific addresses are configured when the zone is booted.

5 **For the network for each interface, add entries to the `/etc/netmasks` file.**

```
## /etc/netmasks in global zone
192.168.2.0 255.255.255.0
192.168.3.0 255.255.255.0
```

For more information, see the `netmasks(4)` man page.

6 **Assign a security template to each zone-specific network interface.**

Create security templates of host type `cipso` that reflect the label of every network. To create and assign the templates, see "Configuring Trusted Network Databases (Task Map)" on page 270.

7 **Run the `txzonemgr` script, and open a separate terminal window.**

In the Labeled Zone Manager, you will add the network interfaces for the labeled zones. In the terminal window, you will display information about the zone and set the default router.

8 **For every zone to which you are going to add a zone-specific network interface and router, complete the following steps:**

a. **In the terminal window, halt the zone.**

```
# zoneadm -z zone-name halt
```

b. **In the Labeled Zone Manager, do the following:**

i. **Select the zone.**

ii. **Select Add Network.**

iii. **Name the network interface.**

iv. **Type the IP address of the interface.**

v. **In the terminal window, verify the zone configuration.**

```
# zonecfg -z zone-name info net
net:    address: IP-address
        physical: zone-network-interface
        defrouter not specified
```

c. **In the terminal window, configure the default router for the labeled zone's network.**

```
# zonecfg -z zone-name
zonecfg:zone-name > select net address=IP-address
zonecfg:zone-name:net> set defrouter=router-address
zonecfg:zone-name:net> end
zonecfg:zone-name > verify
zonecfg:zone-name > commit
zonecfg:zone-name > exit
#
```

For more information, see the zonecfg(1M) man page and "How to Configure the Zone" in *System Administration Guide: Virtualization Using the Solaris Operating System*.

d. **Boot the labeled zone.**

```
# zoneadm -z zone-name boot
```

e. **In the global zone, verify that the labeled zone has a route to the gateway for the subnet.**

```
# netstat -rn
```

A routing table is displayed. The destination and interface for the labeled zone is different from the entry for the global zone.

9 **To remove the default route, select the zone's IP address, then remove the route.**

```
# zonecfg -z zone-name

zonecfg:zone-name > select net address=zone-IP-address
zonecfg:zone-name:net> remove net defrouter=zone-default-route
zonecfg:zone-name:net>  info net
net:
   address: zone-IP-address
   physical: zone-network-interface
   defrouter not specified
```

Example 4–3 Setting a Default Route for a Labeled Zone

In this example, the administrator routes the Secret zone to a separate physical subnet. Traffic to and from the Secret zone is not routed through the global zone. The administrator uses the Labeled Zone Manager and the `zonecfg` command, then verifies that routing works.

The administrator determines that `qfe1` and `qfe1:0` are not currently in use. and creates a mapping for two labeled zones. `qfe1` is the designated interface for the Secret zone.

```
Interface IP Address    Netmask        Default Router
qfe1      192.168.2.22 255.255.255.0 192.168.2.2
qfe1:0    192.168.3.33 255.255.255.0 192.168.3.3
```

First, the administrator creates the `/etc/hostname.qfe1` file and configures the `/etc/netmasks` file.

```
# touch /etc/hostname.qfe1

# cat /etc/netmasks
## /etc/netmasks in global zone
192.168.2.0 255.255.255.0
```

Then, the administrator plumbs the network interface and verifies that the interface is down.

```
# ifconfig qfe1 plumb
# ifconfig -a
```

Then, in the Solaris Management Console, the administrator creates a security template with a single label, Secret, and assigns the IP address of the interface to the template.

The administrator halts the zone.

```
# zoneadm -z secret halt
```

The administrator runs the `txzonemgr` script to open the Labeled Zone Manager.

```
# /usr/sbin/txzonemgr
```

In the Labeled Zone Manager, the administrator selects the Secret zone, selects Add Network, and then selects a network interface. The administrator closes the Labeled Zone Manager.

On the command line, the administrator selects the zone's IP address, then sets its default route. Before exiting the command, the administrator verifies the route and commits it.

```
# zonecfg -z secret
zonecfg: secret > select net address=192.168.6.22
zonecfg: secret:net> set defrouter=192.168.6.2
zonecfg: secret:net> end
zonecfg: secret > verify
zonecfg: secret > commit
zonecfg: secret > info net
  net:
     address: 192.168.6.22
     physical: qfe1
     defrouter: 192.168.6.2
zonecfg: secret > exit
#
```

The administrator boots the zone.

```
# zoneadm -z secret boot
```

In a separate terminal window in the global zone, the administrator verifies the sending and receiving of packets.

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref     Use  Interface
-------------------- -------------------- ----- ----- ------- ---------
default              192.168.5.15         UG        1    2664 qfe0
192.168.6.2          192.168.6.22         UG        1     240 qfe1
192.168.3.3          192.168.3.33         U         1     183 qfe1:0
127.0.0.1            127.0.0.1            UH        1     380 lo0
...
```

▼ Configure a Name Service Cache in Each Labeled Zone

This procedure enables you to separately configure a name service daemon (`nscd`) in each labeled zone. This configuration supports environments where each zone is connected to a subnetwork that runs at the label of the zone, and the subnetwork has its own name server for that label.

Note – This configuration does not satisfy the criteria for an evaluated configuration. In an evaluated configuration, the `nscd` daemon runs only in the global zone. Doors in each labeled zone connect the zone to the global `nscd` daemon.

Before You Begin You are superuser in the global zone. `root` must not yet be a role. You have successfully completed "Add a Network Interface to Route an Existing Labeled Zone" on page 72.

This configuration requires that you have advanced networking skills. If LDAP is your naming service, you are responsible for establishing the LDAP client connection to each labeled zone. The `nscd` daemon caches the name service information, but does not route it.

1 If you are using LDAP, verify a route to the LDAP server from the labeled zone.

In a terminal window in every labeled zone, run the following command:

zone-name # **netstat -rn**

2 In the global zone, start the Labeled Zone Manager.

```
# /usr/sbin/txzonemgr
```

3 **Select the Configure per-zone name service, and click OK.**

This option is intended to be used once, during initial system configuration.

4 **Configure each zone's `nscd` service.**

For assistance, see the `nscd`(1M) and `nscd.conf`(4) man pages.

5 **Reboot the system.**

6 **For every zone, verify the route and the name service daemon.**

a. **In the Zone Console, list the `nscd` service.**

```
zone-name # svcs -x name-service-cache
svc:/system/name-service-cache:default (name service cache)
 State: online since October 10, 2010  10:10:10 AM PDT
   See: nscd(1M)
   See: /etc/svc/volatile/system-name-service-cache:default.log
Impact: None.
```

b. **Verify the route to the subnetwork.**

```
zone-name # netstat -rn
```

7 **To remove the zone-specific name service daemons, do the following in the global zone:**

a. **Open the Labeled Zone Manager.**

b. **Select Unconfigure per-zone name service, and click OK.**

This selection removes the `nscd` daemon in every labeled zone.

c. **Reboot the system.**

Creating Roles and Users in Trusted Extensions

If you are already using administrative roles, you might want to add a Security Administrator role. For sites that have not yet implemented roles, the procedure for creating them is similar to the procedure in the Solaris OS. Trusted Extensions adds the Security Administrator role and requires the use of the Solaris Management Console to administer a Trusted Extensions domain.

If site security requires two people to create user and role accounts, create custom rights profiles and assign them to roles to enforce *separation of duty*.

Task	Description	For Instructions
Create three rights profiles that are more restrictive than default profiles.	Creates rights profiles to manage users. These profiles are more restrictive than the default profiles that manage users.	"Create Rights Profiles That Enforce Separation of Duty" on page 80
Create a security administrator role.	Creates a security administrator role that handles security-relevant tasks.	"Create the Security Administrator Role in Trusted Extensions" on page 83
Create a system administrator role that cannot set a user password.	Creates a system administrator role and assigns to it a restricted System Administrator rights profile.	"Create a Restricted System Administrator Role" on page 85
Create users to assume the administrative roles.	Creates one or more users who can assume roles.	"Create Users Who Can Assume Roles in Trusted Extensions" on page 86
Verify that the roles can perform their tasks.	Tests the roles in various scenarios.	"Verify That the Trusted Extensions Roles Work" on page 88
Enable users to log in to a labeled zone.	Starts the `zones` service so that regular users can log in.	"Enable Users to Log In to a Labeled Zone" on page 90

▼ Create Rights Profiles That Enforce Separation of Duty

Skip this procedure if separation of duty is not a site security requirement. If your site requires separation of duty, you must create these rights profiles and roles before you populate the LDAP server.

This procedure creates rights profiles that have discrete capabilities to manage users. When you assign these profiles to distinct roles, two roles are required to create and configure users. One role can create users, but cannot assign security attributes. The other role can assign security attributes, but cannot create users. When you log in to the Solaris Management Console in a role that is assigned one of these profiles, only the appropriate tabs and fields are available to the role.

Before You Begin You must be superuser, in the `root` role, or in the Primary Administrator role. When you start this procedure, the Solaris Management Console must be closed.

1 **Create copies of the default rights profiles that affect user configuration.**

   a. **Copy the `prof_attr` file to the `prof_attr.orig` file.**

   b. **Open the `prof_attr` file in the trusted editor.**

   ```
   # /usr/dt/bin/trusted_edit /etc/security/prof_attr
   ```

c. Copy the three rights profiles and rename the copies.

```
System Administrator:::Can perform most non-security...
Custom System Administrator:::Can perform most non-security...

User Security:::Manage passwords...
Custom User Security:::Manage passwords...

User Management:::Manage users, groups, home...
Custom User Management:::Manage users, groups, home...
```

d. Save the changes.

e. Verify the changes.

```
# grep ^Custom  /etc/security/prof_attr
Custom System Administrator:::Can perform most non-security...
Custom User Management:::Manage users, groups, home...
Custom User Security:::Manage passwords...
```

Copying a rights profile rather than modifying it enables you to upgrade the system to a later Solaris release and retain your changes. Because these rights profiles are complex, modifying a copy of the default profile is less prone to error than building the more restrictive profile from scratch.

2 Start the Solaris Management Console.

```
# /usr/sbin/smc &
```

3 Select the This Computer (*this-host*: `Scope=Files, Policy=TSOL`) toolbox.

4 Click System Configuration, then click Users.

You are prompted for your password.

5 Type the appropriate password.

6 Double-click Rights.

7 Modify the Custom User Security rights profile.

You restrict this profile from creating a user.

a. Double-click Custom User Security.

b. **Click the Authorizations tab, then perform the following steps:**

i. **From the Included list, remove the** `Manage Users and Roles` **authorization.**

The following User Accounts rights remain:

```
Audit Controls
Label and Clearance Range
Change Password
View Users and Roles
Modify Extended Security Attributes
```

ii. **Add the Manage Privileges right to the Included list.**

c. **Click OK to save your changes.**

8 **Modify the Custom User Management profile.**

You restrict this profile from setting a password.

a. **Double-click Custom User Management.**

b. **Click the Authorizations tab, then perform the following steps:**

i. **Drag the scrollbar for the Included list to User Accounts.**

ii. **From the Included list, remove the** `Modify Extended Security Attributes` **authorization.**

The following User Accounts rights remain:

```
Manage Users and Roles
View Users and Roles
```

c. **Save your changes.**

9 **Modify the Custom System Administrator rights profile.**

The User Management profile is a supplementary profile in this profile. You prevent the system administrator from setting a password.

a. **Double-click Custom System Administrator.**

b. **Click the Supplementary Rights tab, then perform the following steps:**

i. **Remove the User Management rights profile.**

ii. **Add the Custom User Management rights profile.**

iii. **Move the Custom User Management rights profile above the All rights profile.**

c. **Save your changes.**

Next Steps To prevent the default profiles from being used, see Step 7 in "Verify That the Trusted Extensions Roles Work" on page 88 after you verify that the custom profiles enforce separation of duty.

▼ Create the Security Administrator Role in Trusted Extensions

Role creation in Trusted Extensions is identical to role creation in the Solaris OS. However, in Trusted Extensions, a Security Administrator role is required. To create a local Security Administrator role, you can also use the command-line interface, as in Example 4–4.

Before You Begin You must be superuser, in the `root` role, or in the Primary Administrator role.

To create the role on the network, you must have completed "Configuring the Solaris Management Console for LDAP (Task Map)" on page 113.

1. **Start the Solaris Management Console.**

```
# /usr/sbin/smc &
```

2. **Select the appropriate toolbox.**

   - **To create the role locally, use This Computer (***this-host***:** `Scope=Files, Policy=TSOL`**).**
   - **To create the role in the LDAP service, use This Computer (***ldap-server***:** `Scope=LDAP, Policy=TSOL`**).**

3. **Click System Configuration, then click Users.**

   You are prompted for your password.

4. **Type the appropriate password.**

5. **Double-click Administrative Roles.**

6. **From the Action menu, choose Add Administrative Role.**

7. **Create the Security Administrator role.**

Use the following information as a guide:

- Role name – `secadmin`
- Full name – `Security Administrator`
- Description – Site Security Officer *No proprietary information here*.
- Role ID Number – ≥100
- Role shell – Administrator's Bourne (profile shell)
- Create a role mailing list – Leave the checkbox selected.
- Password and confirm – Assign a password of at least 6 alphanumeric characters.

  The password for the Security Administrator role, and all passwords, must be difficult to guess, thus reducing the chance of an adversary gaining unauthorized access by attempting to guess passwords.

  Note – For all administrative roles, make the account Always Available, and do not set password expiration dates.

- Available and Granted Rights – Information Security, User Security
  - If site security does not require separation of duty, select the Information Security and the default User Security rights profiles.
  - If site security requires separation of duty, select the Information Security and the Custom User Security rights profiles.
- Home Directory Server – *home-directory-server*
- Home Directory Path – /*mount-path*
- Assign Users– This field is automatically filled in when you assign a role to a user.

8 After creating the role, check that the settings are correct.

Select the role, then double-click it.

Review the values in the following fields:

- Available Groups – Add groups if required.
- Trusted Extensions Attributes – Defaults are correct.

  For a single-label system where the labels must not be visible, choose Hide for Label: Show or Hide.
- Audit Excluded and Included – Set audit flags only if the role's audit flags are exceptions to the system settings in the `audit_control` file.

9 To create other roles, use the Security Administrator role as a guide.

For examples, see "How to Create and Assign a Role by Using the GUI" in *System Administration Guide: Security Services*. Give each role a unique ID, and assign to the role the correct rights profile. Possible roles include the following:

- admin Role – `System Administrator` Granted Rights
- primaryadmin Role – `Primary Administrator` Granted Rights
- oper Role – `Operator` Granted Rights

Example 4–4 Using the `roleadd` Command to Create a Local Security Administrator Role

In this example, the `root` user adds the Security Administrator role to the local system by using the `roleadd` command. For details, see the `roleadd`(1M) man page. The `root` user consults Table 1–2 before creating the role. At this site, separation of duty is not required to create a user.

```
# roleadd -c "Local Security Administrator" -d /export/home1 \
-u 110 -P "Information Security,User Security" -K lock_after_retries=no \
-K idletime=5 -K idlecmd=lock -K labelview=showsl \
-K min_label=ADMIN_LOW -K clearance=ADMIN_HIGH secadmin
```

The `root` user provides an initial password for the role.

```
# passwd -r files secadmin
New Password:            <Type password>
Re-enter new Password:     <Retype password>
passwd: password successfully changed for secadmin
#
```

To assign the role to a local user, see Example 4–5.

▼ Create a Restricted System Administrator Role

Skip this procedure if separation of duty is not a site security requirement.

In this procedure, you assign a more restrictive rights profile to the System Administrator role.

Before You Begin You must be superuser, in the `root` role, or in the Primary Administrator role.

You have completed "Create Rights Profiles That Enforce Separation of Duty" on page 80. You are using the same toolbox that you used to create the rights profile.

1. **In the Solaris Management Console, create the System Administrator role.**

   For assistance, see "Create the Security Administrator Role in Trusted Extensions" on page 83.

2. **Assign the Custom System Administrator rights profile to the role.**

3. **Save the changes.**

4 Close the Solaris Management Console.

▼ Create Users Who Can Assume Roles in Trusted Extensions

To create a local user, you can use the command-line interface, as in Example 4–5, instead of the following procedure. Where site security policy permits, you can choose to create a user who can assume more than one administrative role.

For secure user creation, the System Administrator role creates the user, and the Security Administrator role assigns security-relevant attributes, such as a password.

Before You Begin You must be superuser, in the `root` role, in the Security Administrator role, or in the Primary Administrator role. The Security Administrator role has the least amount of privilege that is required for user creation.

The Solaris Management Console is displayed. For details, see "Create the Security Administrator Role in Trusted Extensions" on page 83.

1 Double-click User Accounts in the Solaris Management Console.

2 From the Action menu, choose Add User → Use Wizard.

Caution – The names and IDs of roles and users come from the same pool. Do not use existing names or IDs for the users that you add.

---

3 Follow the online help.

You can also follow the procedures in "How to Add a User With the Solaris Management Console's Users Tool" in *System Administration Guide: Basic Administration*.

4 After creating the user, double-click the created user to modify the settings.

Note – For users who can assume roles, make the user account Always Available, and do not set password expiration dates.

Ensure that the following fields are correctly set:

- Description – No proprietary information here.
- Password and confirm – Assign a password of at least 6 alphanumeric characters.

Note – When the initial setup team chooses a password, the team must select a password that is difficult to guess, thus reducing the chance of an adversary gaining unauthorized access by attempting to guess passwords.

- Account Availability – Always Available.
- Trusted Extensions Attributes – Defaults are correct.

  For a single-label system where the labels must not be visible, choose Hide for Label: Show or Hide.
- Account Usage – Set Idle time and Idle action.

  Lock account – Set to No for any user who can assume a role.

5 **Close the Solaris Management Console.**

6 **Customize the user's environment.**

a. **Assign convenient authorizations.**

After checking your site security policy, you might want to grant your first users the Convenient Authorizations rights profile. With this profile, you can enable users to allocate devices, print PostScript™ files, print without labels, remotely log in, and shut down the system. To create the profile, see "How to Create a Rights Profile for Convenient Authorizations" on page 190.

b. **Customize user initialization files.**

See Chapter 13, "Managing Users, Rights, and Roles in Trusted Extensions (Tasks)."

Also see "Managing Users and Rights With the Solaris Management Console (Task Map)" on page 188.

c. **Create multilabel copy and link files.**

On a multilabel system, users and roles can be set up with files that list user initialization files to be copied or linked to other labels. For more information, see "`.copy_files` and `.link_files` Files" on page 179.

Example 4–5 Using the `useradd` Command to Create a Local User

In this example, the `root` user creates a local user who can assume the Security Administrator role. For details, see the `useradd`(1M) and `atohexlabel`(1M) man pages.

First, the `root` user determines the hexadecimal format of the user's minimum label and clearance label.

```
# atohexlabel public
0x0002-08-08
# atohexlabel -c "confidential restricted"
0x0004-08-78
```

Next, the `root` user consults Table 1–2, and then creates the user.

```
# useradd -c "Local user for Security Admin" -d /export/home1 \
-K  idletime=10 -K idlecmd=logout -K lock_after_retries=no
-K min_label=0x0002-08-08 -K clearance=0x0004-08-78 -K labelview=showsl jandoe
```

Then, the `root` user provides an initial password.

```
# passwd -r files jandoe
New Password:          <Type password>
Re-enter new Password:      <Retype password>
passwd: password successfully changed for jandoe
#
```

Finally, the `root` user adds the Security Administrator role to the user's definition. The role was created in "Create the Security Administrator Role in Trusted Extensions" on page 83.

```
# usermod -R secadmin jandoe
```

▼ Verify That the Trusted Extensions Roles Work

To verify each role, assume the role. Then, perform tasks that only that role can perform.

Before You Begin If you have configured DNS or routing, you must reboot after you create the roles and before you verify that the roles work.

1. **For each role, log in as a user who can assume the role.**

2. **Open the Trusted Path menu.**

   In the following trusted stripe, the user name is `tester`.

   a. **Click your user name in the trusted stripe.**

   b. **From the list of roles that are assigned to you, select a role.**

3. **In the role workspace, start the Solaris Management Console.**

```
$ /usr/sbin/smc &
```

4 **Select the appropriate scope for the role that you are testing.**

5 **Click System Services, and navigate to Users.**

You are prompted for a password.

a. **Type the role password.**

b. **Double-click User Accounts.**

6 **Click a user.**

- The System Administrator role should be able to modify fields under the General, Home Directory, and Group tabs.

  If you configured the roles to enforce separation of duty, then the System Administrator role cannot set the user's initial password.
- The Security Administrator role should be able to modify fields under all tabs.

  If you configured the roles to enforce separation of duty, then the Security Administrator role cannot create a user.
- The Primary Administrator role should be able to modify fields under all tabs.

7 **(Optional) If you are enforcing separation of duty, prevent the default rights profiles from being used.**

Note – When the system is upgraded to a newer version of the Solaris OS, the System Administrator, User Management, and User Security default profiles are replaced.

In the trusted editor, perform one of the following steps:

- **Remove the three rights profiles from the `prof_attr` file.**

  Removal prevents an administrator from viewing or assigning these profiles. Also, remove the `prof_attr.orig` file.

- **Comment out the three rights profiles in the `prof_attr` file.**

  Commenting out the rights profiles prevents these profiles from being viewed in the Solaris Management Console or from being used in commands that manage users. The profiles and their contents can still be viewed in the `prof_attr` file.

- **Type a different description for the three rights profiles in the `prof_attr` file.**

  Edit the `prof_attr` file to change the description field of these rights profiles. For example, you might replace the descriptions with `Do not use this profile`. This change warns an administrator to not use the profile, but does not prevent the profile from being used.

▼ Enable Users to Log In to a Labeled Zone

When the host is rebooted, the association between the devices and the underlying storage must be re-established.

Before You Begin You have created at least one labeled zone. That zone is not being used for cloning.

1. **Reboot the system.**

2. **Log in as the `root` user.**

3. **Restart the zones service.**

```
# svcs zones
STATE          STIME    FMRI
offline        -        svc:/system/zones:default

# svcadm restart svc:/system/zones:default
```

4. **Log out.**

   Regular users can now log in. Their session is in a labeled zone.

Creating Home Directories in Trusted Extensions

In Trusted Extensions, users need access to their home directories at every label at which the users work. To make every home directory available to the user requires that you create a multilevel home directory server, run the automounter on the server, and export the home directories. On the client side, you can run scripts to find the home directory for every zone for each user, or you can have the user log in to the home directory server.

▼ Create the Home Directory Server in Trusted Extensions

Before You Begin You must be superuser, in the `root` role, or in the Primary Administrator role.

1. **Install and configure the home directory server with Trusted Extensions software.**
   - If you are cloning zones, make sure that you use a Solaris ZFS snapshot that has empty home directories.
   - Because users require a home directory at every label that they they can log in to, create every zone that a user can log in to. For example, if you use the default `label_encodings` file, you would create a zone for the `PUBLIC` label.

2 If you are using UFS and not Solaris ZFS, enable the NFS server to serve itself.

a. In the global zone, modify the `automount` entry in the `nsswitch.conf` file.

Use the trusted editor to edit the `/etc/nsswitch.conf` file. For the procedure, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

```
automount: files
```

b. In the global zone, run the `automount` command.

3 For every labeled zone, follow the automount procedure in "How to NFS Mount Files in a Labeled Zone" on page 241. Then, return to this procedure.

4 Verify that the home directories have been created.

a. Log out of the home directory server.

b. As a regular user, log in to the home directory server.

c. In the login zone, open a terminal.

d. In the terminal window, verify that the user's home directory exists.

e. Create workspaces for every zone that the user can work in.

f. In each zone, open a terminal window to verify that the user's home directory exists.

5 Log out of the home directory server.

▼ Enable Users to Access Their Home Directories in Trusted Extensions

Users can initially log in to the home directory server to create a home directory that can be shared with other systems. To create a home directory at every label, each user must log in to the home directory server at every label.

Alternatively, you, as administrator, can create a script to create a mount point for home directories on each user's home system before the user first logs in. The script creates mount points at every label at which the user is permitted to work.

Before You Begin The home directory server for your Trusted Extensions domain is configured.

- **Choose whether to allow direct login to the server, or whether to run a script.**
  - **Enable users to log in directly to the home directory server.**

    a. **Instruct each user to log in to the home directory server.**

    After successful login, the user must log out.

    b. **Instruct each user to log in again, and this time, to choose a different login label.**

    The user uses the label builder to choose a different login label. After successful login, the user must log out.

    c. **Instruct each user to repeat the login process for every label that the user is permitted to use.**

    d. **Instruct the users to log in from their regular workstation.**

    Their home directory for their default label is available. When a user changes the label of a session or adds a workspace at a different label, the user's home directory for that label is mounted.

  - **Write a script that creates a home directory mount point for every user, and run the script.**

```
#!/bin/sh
#
for zoneroot in ‘/usr/sbin/zoneadm list -p | cut -d ":" -f4‘ ; do
    if [ $zoneroot != / ]; then
        prefix=$zoneroot/root/export

        for j in ‘getent passwd|tr ’ ’ _‘ ; do
            uid=‘echo $j|cut -d ":" -f3‘
            if [ $uid -ge 100 ]; then
                gid=‘echo $j|cut -d ":" -f4‘
                homedir=‘echo $j|cut -d ":" -f6‘
                mkdir -m 711 -p $prefix$homedir
                chown $uid:$gid $prefix$homedir
            fi
        done
    fi
done
```

    a. **From the global zone, run this script on the NFS server.**

    b. **Then, run this script on every multilevel desktop that the user is going to log in to.**

Adding Users and Hosts to an Existing Trusted Network

If you have users who are defined in NIS maps, you can add them to your network.

To add hosts and labels to hosts, see the following procedures:

- To add a host, you use the Computers and Networks tool set in the Solaris Management Console. For details, see "How to Add Hosts to the System's Known Network" on page 277.

  When you add a host to the LDAP server, add all IP addresses that are associated with the host. All-zones addresses, including addresses for labeled zones, must be added to the LDAP server.
- To label a host, see "How to Assign a Security Template to a Host or a Group of Hosts" on page 278.

▼ Add an NIS User to the LDAP Server

Before You Begin You must be superuser, in the `root` role, or in the Primary Administrator role.

1 From the NIS database, gather the information that you need.

a. Create a file from the user's entry in the `aliases` database.

```
% ypcat -k aliases | grep login-name > aliases.name
```

b. Create a file from the user's entry in the `passwd` database.

```
% ypcat -k passwd | grep "Full Name" > passwd.name
```

c. Create a file from the user's entry in the `auto_home_` database.

```
% ypcat -k auto_home | grep login-name > auto_home_label
```

2 Reformat the information for LDAP and Trusted Extensions.

a. Use the `sed` command to reformat the `aliases` entry.

```
% sed 's/ /:/g' aliases.login-name > aliases
```

b. Use the `nawk` command to reformat the `passwd` entry.

```
% nawk -F: '{print $1":x:"$3":"$4":"$5":"$6":"$7}' passwd.name > passwd
```

c. Use the `nawk` command to create a `shadow` entry.

```
% nawk -F: '{print $1":"$2":6445::::::"}' passwd.name > shadow
```

d. Use the `nawk` command to create a `user_attr` entry.

```
% nawk -F: '{print $1":::lock_after_retries=yes-or-no;profiles=user-profile, ...;
labelview=int-or-ext,show-or-hide;min_label=min-label;
clearance=max-label;type=normal;roles=role-name,...;
auths=auth-name,..."}' passwd.name > user_attr
```

3 Copy the modified files to the `/tmp` directory on the LDAP server.

```
# cp aliases auto_home_internal passwd shadow user_attr /tmp/name
```

4 Add the entries in the files in Step 3 to the databases on the LDAP server.

```
# /usr/sbin/ldapaddent -D "cn=directory manager" -w DM-password \
-a simple -f /tmp/name/aliases aliases
# /usr/sbin/ldapaddent -D "cn=directory manager" -w DM-password \
-a simple -f /tmp/name/auto_home_internal auto_home_internal
# /usr/sbin/ldapaddent -D "cn=directory manager" -w DM-password \
-a simple -f /tmp/name/passwd passwd
# /usr/sbin/ldapaddent -D "cn=directory manager" -w DM-password \
-a simple -f /tmp/name/shadow shadow
# /usr/sbin/ldapaddent -D "cn=directory manager" -w DM-password \
-a simple -f /tmp/name/user_attr user_attr
```

Example 4–6 Adding a User From an NIS Database to the LDAP Server

In the following example, the administrator adds a new user to the trusted network. The user's information is stored originally in an NIS database. To protect the LDAP server password, the administrator runs the `ldapaddent` commands on the server.

In Trusted Extensions, the new user can allocate devices and assume the Operator role. Because the user can assume a role, the user account does not get locked out. The user's minimum label is `PUBLIC`. The label at which the user works is `INTERNAL`, so `jan` is added to the `auto_home_internal` database. The `auto_home_internal` database automounts `jan`'s home directory with read-write permissions.

- On the LDAP server, the administrator extracts user information from NIS databases.

```
# ypcat -k aliases | grep jan.doe > aliases.jan
# ypcat passwd | grep "Jan Doe" > passwd.jan
# ypcat -k auto_home | grep jan.doe > auto_home_internal
```

- Then, the administrator reformats the entries for LDAP.

```
# sed 's/ /:/g' aliases.jan > aliases
# nawk -F: '{print $1":x:"$3":"$4":"$5":"$6":"$7}' passwd.jan > passwd
# nawk -F: '{print $1":"$2":6445::::::"}' passwd.jan > shadow
```

- Then, the administrator creates a `user_attr` entry for Trusted Extensions.

```
# nawk -F: '{print $1"::::lock_after_retries=no;profiles=Media User;
labelview=internal,showsl;min_label=0x0002-08-08;
```

```
clearance=0x0004-08-78;type=normal;roles=oper;
auths=solaris.device.allocate"}' passwd.jan > user_attr
```

- Then, the administrator copies the files to the `/tmp/jan` directory.

```
# cp aliases auto_home_internal passwd shadow user_attr /tmp/jan
```

- Finally, the administrator populates the server with the files in the `/tmp/jan` directory.

```
# /usr/sbin/ldapaddent -D "cn=directory manager" -w a2b3c4d5e6 \
-a simple -f /tmp/jan/aliases aliases
# /usr/sbin/ldapaddent -D "cn=directory manager" -w a2b3c4d5e6 \
-a simple -f /tmp/jan/auto_home_internal auto_home_internal
# /usr/sbin/ldapaddent -D "cn=directory manager" -w a2b3c4d5e6 \
-a simple -f /tmp/jan/passwd passwd
# /usr/sbin/ldapaddent -D "cn=directory manager" -w a2b3c4d5e6 \
-a simple -f /tmp/jan/shadow shadow
# /usr/sbin/ldapaddent -D "cn=directory manager" -w a2b3c4d5e6 \
-a simple -f /tmp/jan/user_attr user_attr
```

Troubleshooting Your Trusted Extensions Configuration

In Trusted Extensions, the labeled zones communicate with the X server through the global zone. Therefore, the labeled zones must have usable routes to the global zone. Also, options that were selected during a Solaris installation can prevent Trusted Extensions from using interfaces to the global zone.

`netservices limited` Was Run After Trusted Extensions Was Enabled

Description:

Instead of running the `netservices limited` command before you enabled Trusted Extensions, you ran the command in the global zone afterwards. Therefore, your labeled zones are unable to connect to the X server in the global zone.

Solution:

Run the following commands to open the services that Trusted Extensions requires to communicate between zones:

```
# svccfg -s x11-server setprop options/tcp_listen = true
# svcadm enable svc:/network/rpc/rstat:default
```

Labeled Zone Is Unable to Access the X Server

Description:

If a labeled zone cannot successfully access the X server, you might see messages such as the following:

- `Action failed. Reconnect to Solaris Zone?`
- `No route available`
- `Cannot reach globalzone-`*hostname*`:0`

Cause:

The labeled zones might not be able to access the X server for any of the following reasons:

- The zone is not initialized and is waiting for the `sysidcfg` process to complete.
- The labeled zone's host name is not recognized by the naming service that runs in the global zone.
- No interface is specified as `all-zones`.
- The labeled zone's network interface is down.
- LDAP name lookups fail.
- NFS mounts do not work.

Steps toward a solution:

Do the following:

1. Log in to the zone.

   You can use the `zlogin` command.

   ```
   # zlogin -z zone-name
   ```

   If you cannot log in as superuser, use the `zlogin -S` command to bypass authentication.
2. Verify that the zone is running.

   ```
   # zoneadm list
   ```

   If a zone has a status of `running`, the zone is running at least one process.
3. Address any problems that prevent the labeled zones from accessing the X server.
   - Initialize the zone by completing the `sysidcfg` process.

     Run the `sysidcfg` program interactively. Answer the prompts in the Zone Terminal Console, or in the terminal window where you ran the `zlogin` command.

To run the `sysidcfg` process noninteractively, you can do one of the following:

- Specify the Initialize item for the `/usr/sbin/txzonemgr` script.

  The Initialize item enables you to supply default values to the `sysidcfg` questions.
- Write your own `sysidcfg` script.

  For more information, see the `sysidcfg(4)` man page.

- Verify that the X server is available to the zone.

  Log in to the labeled zone. Set the `DISPLAY` variable to point to the X server, and open a window.

  ```
  # DISPLAY=global-zone-hostname:n.n
  # export DISPLAY
  # /usr/openwin/bin/xclock
  ```

  If a labeled window does not appear, the zone networking has not been configured correctly for that labeled zone.
- Configure the zone's host name with the naming service.

  The zone's local `/etc/hosts` file is not used. Instead, equivalent information must be specified in the global zone or on the LDAP server. The information must include the IP address of the host name that is assigned to the zone.
- No interface is specified as `all-zones`.

  Unless all your zones have IP addresses on the same subnet as the global zone, you might need to configure an `all-zones` (shared) interface. This configuration enables a labeled zone to connect to the X server of the global zone. If you want to restrict remote connections to the X server of the global zone, you can use `vni0` as the `all-zones` address.

  If you do *not* want an `all-zones` interface configured, you must provide a route to the global zone X server for each zone. These routes must be configured in the global zone.
- The labeled zone's network interface is down.

  ```
  # ifconfig -a
  ```

  Use the `ifconfig` command to verify that the labeled zone's network interface is both `UP` and `RUNNING`.
- LDAP name lookups fail.

  Use the `ldaplist` command to verify that each zone can communicate with the LDAP server or the LDAP proxy server. On the LDAP server, verify that the zone is listed in the `tnrhdb` database.
- NFS mounts do not work.

As superuser, restart `automount` in the zone. Or, add a `crontab` entry to run the `automount` command every five minutes.

Additional Trusted Extensions Configuration Tasks

The following two tasks enable you to transfer exact copies of configuration files to every Trusted Extensions system at your site. The final task enables you to remove Trusted Extensions customizations from a Solaris system.

▼ How to Copy Files to Portable Media in Trusted Extensions

When copying to portable media, label the media with the sensitivity label of the information.

Note – During Trusted Extensions configuration, superuser or an equivalent role copies administrative files to and from portable media. Label the media with `Trusted Path`.

Before You Begin To copy administrative files, you must be superuser or in a role in the global zone.

1. **Allocate the appropriate device.**

   Use the Device Manager, and insert clean media. For details, see "How to Allocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*.

   - In Solaris Trusted Extensions (GNOME), a *File Browser* displays the contents.

   In this procedure, File Browser is used to refer to this GUI.

2. **Open a second File Browser.**

3. **Navigate to the folder that contains the files to be copied**

   For example, you might have copied files to an `/export/clientfiles` folder.

4. **For each file, do the following:**

   a. **Highlight the icon for the file.**

   b. **Drag the file to the File Browser for the portable media.**

5. **Deallocate the device.**

   For details, see "How to Deallocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*.

6 On the File Browser for the portable media, choose Eject from the File menu.

Note – Remember to physically affix a label to the media with the sensitivity label of the copied files.

Example 4–7 Keeping Configuration Files Identical on All Systems

The system administrator wants to ensure that every machine is configured with the same settings. So, on the first machine that is configured, she creates a directory that cannot be deleted between reboots. In that directory, the administrator places the files that should be identical or very similar on all systems.

For example, she copies the Trusted Extensions toolbox that the Solaris Management Console uses for the LDAP scope, `/var/sadm/smc/toolboxes/tsol_ldap/tsol_ldap.tbx`. She has customized remote host templates in the `tnrhtp` file, has a list of DNS servers, and audit configuration files. She also modified the `policy.conf` file for her site. So, she copies the files to the permanent directory.

```
# mkdir /export/commonfiles
# cp  /etc/security/policy.conf \
/etc/security/audit_control \
/etc/security/audit_startup \
/etc/security/tsol/tnrhtp \
/etc/resolv.conf \
/etc/nsswitch.conf \
/export/commonfiles
```

She uses the Device Allocation Manager to allocate a diskette in the global zone, and transfers the files to the diskette. On a separate diskette, labeled `ADMIN_HIGH`, she puts the `label_encodings` file for the site.

When she copies the files onto a system, she modifies the `dir:` entries in the `/etc/security/audit_control` file for that system.

▼ How to Copy Files From Portable Media in Trusted Extensions

It is safe practice to rename the original Trusted Extensions file before replacing the file. When configuring a system, the `root` role renames and copies administrative files.

Before You Begin To copy administrative files, you must be superuser or in a role in the global zone.

1 **Allocate the appropriate device.**

For details, see "How to Allocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*.

In Solaris Trusted Extensions (GNOME), a *File Browser* displays the contents.

2 **Insert the media that contains the administrative files.**

3 **If the system has a file of the same name, copy the original file to a new name.**

For example, add `.orig` to the end of the original file:

```
# cp /etc/security/tsol/tnrhtp /etc/security/tsol/tnrhtp.orig
```

4 **Open a File Browser.**

5 **Navigate to the desired destination directory, such as** `/etc/security/tsol`

6 **For each file that you want to copy, do the following:**

a. **In the File Browser for the mounted media, highlight the icon for the file.**

b. **Then, drag the file to the destination directory in the second File Browser.**

7 **Deallocate the device.**

For details, see "How to Deallocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*.

8 **When prompted, eject and remove the media.**

Example 4–8 Loading Audit Configuration Files in Trusted Extensions

In this example, roles are not yet configured on the system. The `root` user needs to copy configuration files to portable media. The contents of the media will then be copied to other systems. These files are to be copied to each system that is configured with Trusted Extensions software.

The `root` user allocates the `floppy_0` device in the Device Allocation Manager and responds `yes` to the mount query. Then, the `root` user inserts the diskette with the configuration files and copies them to the disk. The diskette is labeled `Trusted Path`.

To read from the media, the `root` user allocates the device on the receiving host, then downloads the contents.

If the configuration files are on a tape, the `root` user allocates the `mag_0` device. If the configuration files are on a CD-ROM, the `root` user allocates the `cdrom_0` device.

How to Remove Trusted Extensions From the System

To remove Trusted Extensions from your Solaris system, you perform specific steps to remove Trusted Extensions customizations to the Solaris system.

1. **As in the Solaris OS, archive any data in the labeled zones that you want to keep.**

2. **Remove the labeled zones from the system.**

   For details, see "How to Remove a Non-Global Zone" in *System Administration Guide: Virtualization Using the Solaris Operating System*.

3. **Disable the Trusted Extensions service.**

   ```
   # svcadm disable labeld
   ```

4. **Run the `bsmunconv` command.**

   For the effect of this command, see the `bsmunconv(1M)` man page.

5. **(Optional) Reboot the system.**

6. **Configure the system.**

   Various services might need to be configured for your Solaris system. Candidates include auditing, basic networking, naming services, and file system mounts.

◆ ◆ ◆ CHAPTER 5

Configuring LDAP for Trusted Extensions (Tasks)

This chapter covers how to configure the Sun Java™ System Directory Server and the Solaris Management Console for use with Solaris Trusted Extensions. The Directory Server provides LDAP services. LDAP is the supported naming service for Trusted Extensions. The Solaris Management Console is the administrative GUI for local and LDAP databases.

You have two options when configuring the Directory Server. You can configure an LDAP server on a Trusted Extensions system, or you can use an existing server and connect to it by using a Trusted Extensions proxy server. Follow the instructions in *one* of the following task maps:

- "Configuring an LDAP Server on a Trusted Extensions Host (Task Map)" on page 103
- "Configuring an LDAP Proxy Server on a Trusted Extensions Host (Task Map)" on page 104

Configuring an LDAP Server on a Trusted Extensions Host (Task Map)

Task	Description	For Instructions
Set up a Trusted Extensions LDAP server.	If you do not have an existing Sun Java System Directory Server, make your first Trusted Extensions system the Directory Server. The other Trusted Extensions systems are clients of this server.	"Collect Information for the Directory Server for LDAP" on page 105 "Install the Sun Java System Directory Server" on page 106 "Configure the Logs for the Sun Java System Directory Server" on page 107
Add Trusted Extensions databases to the server.	Populate the LDAP server with data from the Trusted Extensions system files.	"Populate the Sun Java System Directory Server" on page 110

Task	Description	For Instructions
Configure the Solaris Management Console to work with the Directory Server.	Manually set up an LDAP toolbox for the Solaris Management Console. The toolbox can be used to modify Trusted Extensions attributes on network objects.	"Configuring the Solaris Management Console for LDAP (Task Map)" on page 113
Configure all other Trusted Extensions systems as clients of this server.	When you configure another system with Trusted Extensions, make the system a client of this LDAP server.	"Make the Global Zone an LDAP Client in Trusted Extensions" on page 63

Configuring an LDAP Proxy Server on a Trusted Extensions Host (Task Map)

Use this task map if you have an existing Sun Java System Directory Server that is running on a Solaris system.

Task	Description	For Instructions
Add Trusted Extensions databases to the server.	The Trusted Extensions network databases, `tnrhdb` and `tnrhtp`, need to be added to the LDAP server.	"Populate the Sun Java System Directory Server" on page 110
Set up an LDAP proxy server.	Make one Trusted Extensions system the proxy server for the other Trusted Extensions systems. The other Trusted Extensions systems use this proxy server to reach the LDAP server.	"Create an LDAP Proxy Server" on page 112
Configure the proxy server to have a multilevel port for LDAP.	Enable the Trusted Extensions proxy server to communicate with the LDAP server at specific labels.	"Configure a Multilevel Port for the Sun Java System Directory Server" on page 109
Configure the Solaris Management Console to work with the LDAP proxy server.	You manually set up an LDAP toolbox for the Solaris Management Console. The toolbox can be used to modify Trusted Extensions attributes on network objects.	"Configuring the Solaris Management Console for LDAP (Task Map)" on page 113
Configure all other Trusted Extensions systems as clients of the LDAP proxy server.	When you configure another system with Trusted Extensions, make the system a client of the LDAP proxy server.	"Make the Global Zone an LDAP Client in Trusted Extensions" on page 63

Configuring the Sun Java System Directory Server on a Trusted Extensions System

The LDAP naming service is the supported naming service for Trusted Extensions. If your site is not yet running the LDAP naming service, configure a Sun Java System Directory Server (Directory Server) on a system that is configured with Trusted Extensions. If your site is already running a Directory Server, then you need to add the Trusted Extensions databases to the server. To access the Directory Server, you then set up an LDAP proxy on a Trusted Extensions system.

Note – If you do not use this LDAP server as an NFS server or as a server for Sun Ray™ clients, then you do not need to install any labeled zones on this server.

▼ Collect Information for the Directory Server for LDAP

● **Determine the values for the following items.**

The items are listed in the order of their appearance in the Sun Java Enterprise System Install Wizard.

Install Wizard Prompt	Action or Information
Sun Java System Directory Server *version*	
Administrator User ID	The default value is `admin`.
Administrator Password	Create a password, such as `admin123`.
Directory Manager DN	The default value is `cn=Directory Manager`.
Directory Manager Password	Create a password, such as `dirmgr89`.
Directory Server Root	The default value is `/var/Sun/mps`. This path is also used later if the proxy software is installed.
Server Identifier	The default value is the local system.
Server Port	If you plan to use the Directory Server to provide standard LDAP naming services to client systems, use the default value, `389`. If you plan to use the Directory Server to support a subsequent installation of a proxy server, enter a nonstandard port, such as `10389`.
Suffix	Include your domain component, as in `dc=example-domain,dc=com`.
Administration Domain	Construct to correspond to the Suffix, as in, `example-domain.com`.

Install Wizard Prompt	Action or Information
System User	The default value is `root`.
System Group	The default value is `root`.
Data Storage Location	The default value is `Store configuration data on this server`.
Data Storage Location	The default value is `Store user data and group data on this server`.
Administration Port	The default value is the Server Port. A suggested convention for changing the default is `software-version TIMES 1000`. For software version 5.2, this convention would result in port `5200`.

▼ Install the Sun Java System Directory Server

The Directory Server packages are available from the Sun Software Gateway web site (`http://www.sun.com/software/solaris`).

1 Find the Sun Java System Directory Server packages on the Sun web site.

a. On the Sun Software Gateway (`http://www.sun.com/software/solaris`) page, click the Get It tab.

b. Click the checkbox for the Sun Java Identity Management Suite.

c. Click the Submit button.

d. If you are not registered, register.

e. Log in to download the software.

f. Click the Download Center at the upper left of the screen.

g. Under Identity Management, download the most recent software that is appropriate for your platform.

2 In the `/etc/hosts` file, add the FQDN to your system's hostname entry.

The FQDN is the Fully Qualified Domain Name. This name is a combination of the host name and the administration domain, as in:

```
192.168.5.5 myhost myhost.example-domain.com
```

3 Install the Directory Server packages.

Answer the questions by using the information from "Collect Information for the Directory Server for LDAP" on page 105.

4 Ensure that the Directory Server starts at every boot.

Templates for the SMF services for the Directory Server are in the Sun Java System Directory Server packages.

- **For a Trusted Extensions Directory Server, enable the service.**

  ```
  # dsadm stop /export/home/ds/instances/your-instance
  # dsadm enable-service -T SMF /export/home/ds/instances/your-instance
  # dsadm start /export/home/ds/instances/your-instance
  ```

  For information about the `dsadm` command, see the `dsadm`(1M) man page.

- **For a proxy Directory Server, enable the service.**

  ```
  # dpadm stop /export/home/ds/instances/your-instance
  # dpadm enable-service -T SMF /export/home/ds/instances/your-instance
  # dpadm start /export/home/ds/instances/your-instance
  ```

  For information about the `dpadm` command, see the `dpadm`(1M) man page.

5 Verify your installation.

```
# dsadm info /export/home/ds/instances/your-instance
Instance Path:         /export/home/ds/instances/your-instance
Owner:                 root(root)
Non-secure port:       389
Secure port:           636
Bit format:            32-bit
State:                 Running
Server PID:            298
DSCC url:              -
SMF application name:  ds--export-home-ds-instances-your-instance
Instance version:      D-A00
```

Troubleshooting For strategies to solve LDAP configuration problems, see Chapter 13, "LDAP Troubleshooting (Reference)," in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

▼ Configure the Logs for the Sun Java System Directory Server

This procedure configures three types of logs: access logs, audit logs, and error logs. The following default settings are not changed:

- All logs are enabled and buffered.
- Logs are placed in the appropriate `/export/home/ds/instances/`*your-instance*`/logs/`*LOG_TYPE* directory.

- Events are logged at log level 256.
- Logs are protected with `600` file permissions.
- Access logs are rotated daily.
- Error logs are rotated weekly.

The settings in this procedure meet the following requirements:

- Audit logs are rotated daily.
- Log files that are older than 3 months expire.
- All log files use a maximum of 20,000 MBytes of disk space.
- A maximum of 100 log files is kept, and each file is at most 500 MBytes.
- The oldest logs are deleted if less than 500 MBytes free disk space is available.
- Additional information is collected in the error logs.

1 Configure the access logs.

The *LOG_TYPE* for access is `ACCESS`. The syntax for configuring logs is the following:

`dsconf set-log-prop` *LOG_TYPE property*`:`*value*

```
# dsconf set-log-prop ACCESS max-age:3M
# dsconf set-log-prop ACCESS max-disk-space-size:20000M
# dsconf set-log-prop ACCESS max-file-count:100
# dsconf set-log-prop ACCESS max-size:500M
# dsconf set-log-prop ACCESS min-free-disk-space:500M
```

2 Configure the audit logs.

```
# dsconf set-log-prop AUDIT max-age:3M
# dsconf set-log-prop AUDIT max-disk-space-size:20000M
# dsconf set-log-prop AUDIT max-file-count:100
# dsconf set-log-prop AUDIT max-size:500M
# dsconf set-log-prop AUDIT min-free-disk-space:500M
# dsconf set-log-prop AUDIT rotation-interval:1d
```

By default, the rotation interval for audit logs is one week.

3 Configure the error logs.

In this configuration, you specify additional data to be collected in the error log.

```
# dsconf set-log-prop ERROR max-age:3M
# dsconf set-log-prop ERROR max-disk-space-size:20000M
# dsconf set-log-prop ERROR max-file-count:30
# dsconf set-log-prop ERROR max-size:500M
# dsconf set-log-prop ERROR min-free-disk-space:500M
# dsconf set-log-prop ERROR verbose-enabled:on
```

4 (Optional) Further configure the logs.

You can also configure the following settings for each log:

```
# dsconf set-log-prop LOG_TYPE rotation-min-file-size:undefined
# dsconf set-log-prop LOG_TYPE rotation-time:undefined
```

For information about the `dsconf` command, see the `dsconf`(1M) man page.

▼ Configure a Multilevel Port for the Sun Java System Directory Server

To work in Trusted Extensions, the server port of the Directory Server must be configured as a multilevel port (MLP) in the global zone.

1 Start the Solaris Management Console.

```
# /usr/sbin/smc &
```

2 Select the This Computer (*this-host*: `Scope=Files, Policy=TSOL`) toolbox.

3 Click System Configuration, then click Computers and Networks.

You are prompted for your password.

4 Type the appropriate password.

5 Double-click Trusted Network Zones.

6 Double-click the global zone.

7 Add a multilevel port for the TCP protocol:

a. Click Add for the Multilevel Ports for Zone's IP Addresses.

b. Type 389 for the port number, and click OK.

8 Add a multilevel port for the UDP protocol:

a. Click Add for the Multilevel Ports for Zone's IP Addresses.

b. Type 389 for the port number.

c. Choose the udp protocol, and click OK.

9 Click OK to save the settings.

10 **Update the kernel.**

```
# tnctl -fz /etc/security/tsol/tnzonecfg
```

Populate the Sun Java System Directory Server

Several LDAP databases have been created or modified to hold Trusted Extensions data about label configuration, users, and remote systems. In this procedure, you populate the Directory Server databases with Trusted Extensions information.

Before You Begin If site security requires separation of duty, complete the following before populating the Directory server:

- "Create Rights Profiles That Enforce Separation of Duty" on page 80
- "Create the Security Administrator Role in Trusted Extensions" on page 83
- "Create a Restricted System Administrator Role" on page 85

1 **Create a staging area for files that you plan to use to populate the naming service databases.**

```
# mkdir -p /setup/files
```

2 **Copy the sample `/etc` files into the staging area.**

```
# cd /etc
# cp aliases group networks netmasks protocols /setup/files
# cp rpc services auto_master /setup/files

# cd /etc/security
# cp auth_attr prof_attr exec_attr /setup/files/
#
# cd /etc/security/tsol
# cp tnrhdb tnrhtp /setup/files

# cd /etc/inet
# cp ipnodes /setup/files
```

3 **Remove the `+auto_master` entry from the `/setup/files/auto_master` file.**

4 **Remove the `?:::::?` entry from the `/setup/files/auth_attr` file.**

5 **Remove the `::::` entry from the `/setup/files/prof_attr` file.**

6 **Create the zone automaps in the staging area.**

In the following list of automaps, the first of each pair of lines shows the name of the file. The second line of each pair shows the file contents. The zone names identify labels from the default `label_encodings` file that is included with the Trusted Extensions software.

- Substitute your zone names for the zone names in these lines.
- *myNFSserver* identifies the NFS server for the home directories.

```
/setup/files/auto_home_public
 * myNFSserver_FQDN:/zone/public/root/export/home/&

/setup/files/auto_home_internal
 * myNFSserver_FQDN:/zone/internal/root/export/home/&

/setup/files/auto_home_needtoknow
 * myNFSserver_FQDN:/zone/needtoknow/root/export/home/&

/setup/files/auto_home_restricted
 * myNFSserver_FQDN:/zone/restricted/root/export/home/&
```

7 **Add every system on the network to the `/setup/files/tnrhdb` file.**

No wildcard mechanism can be used here. The IP address of every system to be contacted, including the IP addresses of labeled zones, *must* be in this file.

a. **Open the trusted editor and edit `/setup/files/tnrhdb`.**

b. **Add every IP address on a labeled system in the Trusted Extensions domain.**

Labeled systems are of type `cipso`. Also, the name of the security template for labeled systems is `cipso`. Therefore, in the default configuration, a `cipso` entry is similar to the following:

```
192.168.25.2:cipso
```

Note – This list includes the IP addresses of global zones and labeled zones.

---

c. **Add every unlabeled system with which the domain can communicate.**

Unlabeled systems are of type `unlabeled`. The name of the security template for unlabeled systems is `admin_low`. Therefore, in the default configuration, an entry for an unlabeled system is similar to the following:

```
192.168.35.2:admin_low
```

d. **Save the file, and exit the editor.**

e. **Check the syntax of the file.**

```
# tnchkdb -h /setup/files/tnrhdb
```

f. **Fix any errors before continuing.**

8 **Copy the** `/setup/files/tnrhdb` **file to the** `/etc/security/tsol/tnrhdb` **file.**

9 **Use the** `ldapaddent` **command to populate every file in the staging area.**

```
# /usr/sbin/ldapaddent -D "cn=directory manager" \
-w dirmgr123 -a simple -f /setup/files/hosts hosts
```

Creating a Trusted Extensions Proxy for an Existing Sun Java System Directory Server

First, you need to add the Trusted Extensions databases to the existing Directory Server on a Solaris system. Second, to enable Trusted Extensions systems to access the Directory Server, you then need to configure a Trusted Extensions system to be the LDAP proxy server.

▼ Create an LDAP Proxy Server

If an LDAP server already exists at your site, create a proxy server on a Trusted Extensions system.

Before You Begin You have added the databases that contain Trusted Extensions information to the LDAP server. For details, see "Populate the Sun Java System Directory Server" on page 110.

1 **On a system that is configured with Trusted Extensions, create a proxy server.**

For details, see Chapter 12, "Setting Up LDAP Clients (Tasks)," in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

2 **Verify that the Trusted Extensions databases can be viewed by the proxy server.**

```
# ldaplist -l database
```

Troubleshooting For strategies to solve LDAP configuration problems, see Chapter 13, "LDAP Troubleshooting (Reference)," in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

Configuring the Solaris Management Console for LDAP (Task Map)

The Solaris Management Console is the GUI for administering the network of systems that are running Trusted Extensions.

Task	Description	For Instructions
Initialize the Solaris Management Console.	Initialize the Solaris Management Console. This procedure is performed once per system in the global zone.	"Initialize the Solaris Management Console Server in Trusted Extensions" on page 60
Register credentials.	Authenticate the Solaris Management Console with the LDAP server.	"Register LDAP Credentials With the Solaris Management Console" on page 113
Enable remote administration on a system.	By default, a Solaris Management Console client cannot communicate with a Console server on another system. You must explicitly enable remote administration.	"Enable the Solaris Management Console to Accept Network Communications" on page 114
Create the LDAP toolbox.	Create the LDAP toolbox in the Solaris Management Console for Trusted Extensions.	"Edit the LDAP Toolbox in the Solaris Management Console" on page 115
Verify communications.	Verify that Trusted Extensions hosts can become LDAP clients.	"Verify That the Solaris Management Console Contains Trusted Extensions Information" on page 117

▼ Register LDAP Credentials With the Solaris Management Console

Before You Begin You must be the `root` user on an LDAP server that is running Trusted Extensions. The server can be a proxy server.

Your Sun Java System Directory Server must be configured. You have completed one of the following configurations:

- "Configuring an LDAP Server on a Trusted Extensions Host (Task Map)" on page 103
- "Configuring an LDAP Proxy Server on a Trusted Extensions Host (Task Map)" on page 104

1 Register the LDAP administrative credentials.

```
LDAP-Server # /usr/sadm/bin/dtsetup storeCred
Administrator DN:       Type the value for cn on your system
Password:       Type the Directory Manager password
Password (confirm):       Retype the password
```

2 List the scopes on the Directory Server.

```
LDAP-Server # /usr/sadm/bin/dtsetup scopes
Getting list of manageable scopes...
Scope 1 file:     Displays name of file scope
Scope 2 ldap:     Displays name of ldap scope
```

Your LDAP server setup determines the scopes that are listed. The LDAP scope is not listed until the LDAP toolbox is edited. The toolbox cannot be edited until after the server is registered.

Example 5–1 Registering LDAP Credentials

In this example, the name of the LDAP server is `LDAP1` and the value for `cn` is the default, `Directory Manager`.

```
# /usr/sadm/bin/dtsetup storeCred
Administrator DN:cn=Directory Manager
Password:abcde1;!
Password (confirm):abcde1;!
# /usr/sadm/bin/dtsetup scopes
Getting list of manageable scopes...
Scope 1 file:/LDAP1/LDAP1
Scope 2 ldap:/LDAP1/cd=LDAP1,dc=example,dc=com
```

▼ Enable the Solaris Management Console to Accept Network Communications

By default, Solaris systems are not configured to listen on ports that present security risks. Therefore, you must explicitly configure any system that you plan to administer remotely to accept network communications. For example, to administer network databases on the LDAP server from a client, the Solaris Management Console server on the LDAP server must accept network communications.

For an illustration of the Solaris Management Console configuration requirements for a network with an LDAP server, see "Client-Server Communication With the Solaris Management Console" on page 144.

Before You Begin You must be superuser in the global zone on the Solaris Management Console server system. In this procedure, that system is called the remote system. Also, you must have command line access to the client system as superuser.

1 On the remote system, enable the system to accept remote connections.

The `smc` daemon is controlled by the `wbem` service. If the `options/tcp_listen` property to the `wbem` service is set to `true`, the Solaris Management Console server accepts remote connections.

```
# /usr/sbin/svcprop -p options wbem
options/tcp_listen boolean false
# svccfg -s wbem setprop options/tcp_listen=true
```

2 Refresh and restart the `wbem` service.

```
# svcadm refresh wbem
# svcadm restart wbem
```

3 Verify that the `wbem` service is set to accept remote connections.

```
# svcprop -p options wbem
options/tcp_listen boolean true
```

4 On the remote system and on any client that needs to access the Solaris Management Console, ensure that remote connections are enabled in the `smcserver.config` file.

a. Open the `smcserver.config` file in the trusted editor.

```
# /usr/dt/bin/trusted_edit /etc/smc/smcserver.config
```

b. Set the `remote.connections` parameter to `true`.

```
## remote.connections=false
remote.connections=true
```

c. Save the file and exit the trusted editor.

Troubleshooting If you restart or enable the `wbem` service, you must ensure that the `remote.connections` parameter in the `smcserver.config` file remains set to `true`.

▼ Edit the LDAP Toolbox in the Solaris Management Console

Before You Begin You must be superuser on the LDAP server. The LDAP credentials must be registered with the Solaris Management Console, and you must know the output of the `/usr/sadm/bin/dtsetup scopes` command. For details, see "Register LDAP Credentials With the Solaris Management Console" on page 113.

1 Find the LDAP toolbox.

```
# cd /var/sadm/smc/toolboxes/tsol_ldap
# ls *tbx
tsol_ldap.tbx
```

2 Provide the LDAP server name.

a. Open the trusted editor.

b. Copy and paste the full pathname of the `tsol_ldap.tbx` toolbox as the argument to the editor.

For example, the following path is the default location of the LDAP toolbox:

```
/var/sadm/smc/toolboxes/tsol_ldap/tsol_ldap.tbx
```

c. Replace the scope information.

Replace the `server` tags between the `<Scope>` and `</Scope>` tags with the output of the `ldap:/......` line from the `/usr/sadm/bin/dtsetup scopes` command.

```
<Scope>ldap:/<ldap-server-name>/<dc=domain,dc=suffix></Scope>
```

d. Replace every instance of `<?server?>` or `<?server ?>` with the LDAP server.

```
<Name>This Computer (ldap-server-name: Scope=ldap, Policy=TSOL)</Name>
services and configuration of ldap-server-name.</Description>
and configuring ldap-server-name.</Description>
...
```

e. Save the file, and exit the editor.

3 Refresh and restart the `wbem` service.

```
# svcadm refresh wbem
# svcadm restart wbem
```

Example 5–2 Configuring the LDAP Toolbox

In this example, the name of the LDAP server is `LDAP1`. To configure the toolbox, the administrator replaces the instances of `<?server ?>` with `LDAP1`.

```
# cd /var/sadm/smc/toolboxes/tsol_ldap
# /usr/dt/bin/trusted_edit /tsol_ldap.tbx
<Scope>ldap:/LDAP1/cd=LDAP1,dc=example,dc=com</Scope

...
<Name>This Computer (LDAP1: Scope=ldap, Policy=TSOL)</Name>
services and configuration of LDAP1.</Description>
and configuring LDAP1.</Description>
...
```

▼ Verify That the Solaris Management Console Contains Trusted Extensions Information

For an illustration of the Solaris Management Console configuration requirements for a network with an LDAP server and for a network without an LDAP server, see "Client-Server Communication With the Solaris Management Console" on page 144.

Before You Begin You must be logged in to an LDAP client in an administrative role, or as superuser. To make a system an LDAP client, see "Make the Global Zone an LDAP Client in Trusted Extensions" on page 63.

To administer the local system, you must have completed "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

To connect to a Console server on a remote system from the local system, you must have completed "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60 on both systems. Also, on the remote system, you must have completed "Enable the Solaris Management Console to Accept Network Communications" on page 114.

To administer the databases in the LDAP naming service from the LDAP client, on the LDAP server you must have completed "Edit the LDAP Toolbox in the Solaris Management Console" on page 115, in addition to the preceding procedures.

1. **Start the Solaris Management Console.**

```
# /usr/sbin/smc &
```

2. **Open a Trusted Extensions toolbox.**

   A Trusted Extensions toolbox has the value `Policy=TSOL`.

   - **On a trusted network that uses LDAP as a naming service, perform the following tests:**

     a. **To check that local administrative databases can be accessed, open the following toolbox:**

        `This Computer (`*this-host*`: Scope=Files, Policy=TSOL)`

     b. **To check that the LDAP server's local administrative databases can be accessed, specify the following toolbox:**

        `This Computer (`*ldap-server*`: Scope=Files, Policy=TSOL)`

     c. **To check that the naming service databases on the LDAP server can be accessed, specify the following toolbox:**

        `This Computer (`*ldap-server*`: Scope=LDAP, Policy=TSOL)`

- **On a trusted network that does not use LDAP as a naming service, perform the following tests:**

  a. **To check that local administrative databases can be accessed, open the following toolbox:**

     `This Computer (`*this-host*`: Scope=Files, Policy=TSOL)`

  b. **To check that a remote system's local administrative databases can be accessed, specify the following toolbox:**

     `This Computer (`*remote-system*`: Scope=Files, Policy=TSOL)`

3 Under System Configuration, navigate to Computers and Networks, then Security Templates.

4 Check that the correct templates and labels have been applied to the remote systems.

Note – When you try to access network database information from a system that is not the LDAP server, the operation fails. The Console allows you to log in to the remote host and open the toolbox. However, when you try to access or change information, the following error message indicates that you have selected `Scope=LDAP` on a system that is not the LDAP server:

```
Management server cannot perform the operation requested.
...
Error extracting the value-from-tool.
The keys received from the client were machine, domain, Scope.
Problem with Scope.
```

Troubleshooting To troubleshoot LDAP configuration, see Chapter 13, "LDAP Troubleshooting (Reference)," in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

CHAPTER 6

Configuring a Headless System With Trusted Extensions (Tasks)

Configuring and administering Solaris Trusted Extensions software on headless systems such as the Netra™ series requires modifying security settings on the headless system to enable remote access. Administering a remote Trusted Extensions system requires similar setup. To run an administrative GUI, you might need to run the process on the remote system and display the GUI on the desktop system.

For an explanation of the requirements, see Chapter 14, "Remote Administration in Trusted Extensions (Tasks)"

Note – The configuration methods that headless and remote systems require do not satisfy the criteria for an evaluated configuration. For more information, see "Understanding Your Site's Security Policy" on page 28.

Headless System Configuration in Trusted Extensions (Task Map)

On headless systems, a console is connected by means of a serial line to a terminal emulator window. The line is typically secured by the `tip` command. Depending on what type of second system is available, you can use one of the following methods to configure a headless system. The methods are listed from more secure to less secure in the following table. These instructions also apply to remote systems.

Task	Description	For Instructions
Enable remote login by the `root` user.	If you are not using LDAP, you must initially log in to the headless system as `root`. If you are using LDAP, you can skip this procedure.	"Enable Remote Login by `root` User in Trusted Extensions" on page 120

Task	Description	For Instructions
Enable remote login.	Enable remote login for a user who can assume the `root` role or another administrative role.	"Enable Remote Login by a Role in Trusted Extensions" on page 121
	Enable the administration of Trusted Extensions systems from an unlabeled system.	"Enable Remote Login From an Unlabeled System" on page 123
	Enable a user to access the global zone on a headless system.	"How to Enable Specific Users to Log In Remotely to the Global Zone in Trusted Extensions" on page 206
(Optional) Enable the display of administrative GUIs.	Enable administrative GUIs that run on the headless system to display on the desktop system.	"Enable the Remote Display of Administrative GUIs" on page 124
(Optional) Enable virtual network computing (vnc)	From any client, uses the Xvnc server on the remote Trusted Extensions to display a multilevel session back to the client.	"How to Use Xvnc to Remotely Access a Trusted Extensions System" on page 207
Choose a configuration and administration method to set up the headless system.	Assume a role or become superuser to administer the remote system.	"Use the `rlogin` or `ssh` Command to Log In and Administer a Headless System in Trusted Extensions" on page 125
	Use the Solaris Management Console on the headless system.	"Use a Remote Solaris Management Console to Administer in the Files Scope" on page 123
	If you have no windowing system, you can use serial login as superuser. This procedure is insecure.	No configuration is required.

Note – Consult your security policy to determine which methods of remote administration are permissible at your site.

▼ Enable Remote Login by `root` User in Trusted Extensions

As in the Solaris OS, `root` can log in remotely from a labeled system when the `CONSOLE` entry is disabled.

If you plan to administer a remote system by editing local files, use this procedure.

1 **In the trusted editor, comment out the `CONSOLE=` line in the `/etc/default/login` file.**

```
# /usr/dt/bin/trusted_edit /etc/default/login
```

The edited line appears similar to the following:

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

2 Permit `root` user login over an `ssh` connection.

Modify the `/etc/ssh/sshd_config` file. By default, `ssh` is enabled on a Solaris system.

```
# /usr/dt/bin/trusted_edit /etc/ssh/sshd_config
```

The edited line appears similar to the following:

```
PermitRootLogin yes
```

Next Steps To log in as the `root` user from an unlabeled system, you must also complete "Enable Remote Login From an Unlabeled System" on page 123.

To enable remote login by a role, continue with "Enable Remote Login by a Role in Trusted Extensions" on page 121.

▼ Enable Remote Login by a Role in Trusted Extensions

Follow this procedure *only if* you must administer a headless system by using the `rlogin` or `ssh` command.

Configuration errors can be debugged remotely.

Before You Begin If you are using local files to administer the remote system, you have completed "Enable Remote Login by `root` User in Trusted Extensions" on page 120. Then, as the `root` user, perform this task on both systems.

1 On both systems, identify the other system as a labeled system.

The desktop system and the headless system must identify each other as using the identical security template. For the procedure, see "How to Assign a Security Template to a Host or a Group of Hosts" on page 278.

To assign a temporary label, see Example 6–1.

2 On both systems, create identical users and roles.

The names and IDs must be identical, and the role must be assigned to the user on both systems. To create users and roles, see "Creating Roles and Users in Trusted Extensions" on page 79.

3 To contact a remote Solaris Management Console, do the following on both systems:

a. Add the other system's host name and IP address to the `/etc/hosts` file.

```
# /usr/dt/bin/trusted_edit /etc/hosts

127.0.0.1    localhost
192.168.66.66    local-system-name    loghost
192.168.66.12    remote-system-name
```

b. **To allow remote role assumption, modify the `pam.conf` file to relax PAM policy.**

   i. **Copy the `/etc/pam.conf` file to `/etc/pam.conf.orig`.**

   ```
   # cp /etc/pam.conf /etc/pam.conf.orig
   ```

   ii. **In the trusted editor, open the `pam.conf` file.**

   ```
   # /usr/dt/bin/trusted_edit /etc/pam.conf
   ```

   iii. **Copy the default entries under Account management.**

   iv. **In each copied entry, change `other` to `smcconsole`.**

   v. **To the copied `pam_roles.so.1` entry, add `allow_remote`.**

   Use the Tab key between fields. This section now appears similar to the following:

   ```
   # Solaris Management Console definition for Account management
   #
   smcconsole   account requisite   pam_roles.so.1   allow_remote
   smcconsole   account required    pam_unix_account.so.1
   smcconsole   account required    pam_tsol_account.so.1

   # Default definition for Account management
   # Used when service name is not explicitly mentioned for account management
   #
   other   account requisite    pam_roles.so.1
   other   account required     pam_unix_account.so.1
   other   account required     pam_tsol_account.so.1
   ```

   vi. **Save the file and exit the editor.**

   vii. **(Optional) Copy the file to `/etc/pam.conf.site`.**

   ```
   # cp /etc/pam.conf /etc/pam.conf.site
   ```

   If you upgrade the system to a later release, you must then evaluate if you should copy the changes from `/etc/pam.conf.site` into the `pam.conf` file.

Example 6–1 Creating a Temporary Definition of a Trusted Extensions Host Type

In this example, the administrator wants to start configuring a remote Trusted Extensions system before the host type definitions are set up. To do so, the administrator uses the `tnctl` command on the remote system to temporarily define the host type of the desktop system:

```
remote-TX# tnctl -h desktop-TX:cipso
```

Later, the administrator wants to reach the remote Trusted Extensions system from a desktop system that is not configured with Trusted Extensions. In this case, the administrator uses the `tnctl` command on the remote system to temporarily define the host type of the desktop system as an unlabeled system that runs at the `ADMIN_LOW` label:

```
remote-TX# tnctl -h desktop-TX:admin_low
```

Enable Remote Login From an Unlabeled System

Before You Begin This procedure is not secure.

You have relaxed PAM policy to allow remote role assumption, as described in "Enable Remote Login by a Role in Trusted Extensions" on page 121.

1. **On the trusted system, apply the appropriate security template to the unlabeled system.**

Caution – With the default settings, another unlabeled system could log in and administer the remote system. Therefore, you must change the `0.0.0.0` network default from `ADMIN_LOW` to a different label. For the procedure, see "How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 279.

2. **In the trusted editor, open the `/etc/pam.conf` file.**

```
# /usr/dt/bin/trusted_edit /etc/pam.conf
```

3. **Find the `smcconsole` entries.**

4. **Add `allow_unlabeled` to the `tsol_account` module.**

Use the Tab key between fields.

```
smcconsole   account required  pam_tsol_account.so.1 allow_unlabeled
```

After your edits, this section appears similar to the following:

```
# Solaris Management Console definition for Account management
#
smcconsole  account  requisite      pam_roles.so.1     allow_remote
smcconsole  account  required       pam_unix_account.so.1
smcconsole  account  required       pam_tsol_account.so.1 allow_unlabeled
```

Use a Remote Solaris Management Console to Administer in the Files Scope

If you are not using LDAP, and you want to use the Solaris Management Console on a remote system, you enable remote connection to the Console. This procedure is not sufficient to enable access for the LDAP scope.

To enable access for the LDAP scope, you must complete all the procedures in “Configuring the Solaris Management Console for LDAP (Task Map)” on page 113.

Before You Begin Both systems are labeled systems.

You have completed the following procedures:

- “Initialize the Solaris Management Console Server in Trusted Extensions” on page 60
- “Enable Remote Login by a Role in Trusted Extensions” on page 121

1. **Complete “Enable the Solaris Management Console to Accept Network Communications” on page 114.**

2. **On the desktop system, become a user that is defined identically on both systems.**

3. **On the desktop system, assume the role that is defined identically on both systems.**

4. **On the desktop system, start the Solaris Management Console.**

```
# /usr/sbin/smc &
```

5. **In the Server dialog box, type the name of the headless system.**

   Then, choose the `Scope=Files` toolbox.

```
This Computer (remote-system: Scope=Files, Policy=TSOL)
```

▼ Enable the Remote Display of Administrative GUIs

The procedure for remote display on a desktop is identical to the procedure on a Solaris system that is not configured with Trusted Extensions. This procedure is placed here for convenience.

1. **On the desktop system, enable processes from the headless system to display.**

   a. **Enable the headless system to access the X server on the desktop system.**

```
desktop $ xhost + headless-host
```

   b. **Determine the value of the desktop's `DISPLAY` variable.**

```
desktop $ echo $DISPLAY
:n.n
```

2. **On the headless system, set the `DISPLAY` variable to the desktop system.**

```
headless $ DISPLAY=desktop:n.n
headless $ export DISPLAY=n:n
```

▼ Use the `rlogin` or `ssh` Command to Log In and Administer a Headless System in Trusted Extensions

This procedure enables you to use the command line and the `txzonemgr` GUI to administer a headless system as superuser or as a role.

Note – Remote login by using the `rlogin` command is less secure than remote login by using the `ssh` command.

To use the Solaris Management Console to administer a remote system does not require you to use a remote login command. For the procedure, see "How to Remotely Administer Systems by Using the Solaris Management Console From a Trusted Extensions System" on page 203.

Before You Begin You have completed "Enable Remote Login by a Role in Trusted Extensions" on page 121.

You are a user who is enabled to log in to the headless system with that same user name and user ID, and you can assume the same role on the headless system that you can assume on the desktop system.

1 On the desktop system, enable processes from the headless system to display.

```
desktop $ xhost + headless-host
desktop $ echo $DISPLAY
:n.n
```

2 Ensure that you are the user who is identically defined on both systems.

3 From a terminal window, remotely log in to the headless system.

- **Use the `ssh` command to log in:**

  ```
  desktop $ ssh -l identical-username  headless
  Password:       Type the user's password
  headless $
  ```

- **Or, use the `rlogin` command to log in:**

  ```
  desktop # rlogin headless
  Password:       Type the user's password
  headless $
  ```

4 Assume the role that is defined identically on both systems.

Use the same terminal window. For example, assume the `root` role.

```
headless $ su - root
Password:       Type the root password
```

You are now in the global zone. You can now use this terminal to administer the headless system from the command line.

5 Enable processes on the headless system to display on the desktop system.

Note – You can also display remote GUIs by logging in with the `ssh -X` command. For more information, see the `ssh(1)` man page. For an example, see Example 6–2.

```
headless $  DISPLAY desktop:n.n
headless $ export DISPLAY=n:n
```

You can now administer the headless system by using Trusted Extensions GUIs. For example, start the `txzonemgr` GUI:

```
headless $ /usr/sbin/txzonemgr
```

The Labeled Zone Manager runs on the remote system and displays on the desktop system.

Example 6–2 Configuring Labeled Zones on a Headless System

In this example, the administrator uses the `txzonemgr` GUI to configure labeled zones on a labeled headless system from a labeled desktop system. As in the Solaris OS, the administrator enables X server access to the desktop system by using the `-X` option to the `ssh` command. The user `install1` is defined identically on both systems, and can assume the role `remoterole`.

```
TXdesk1 $ xhost + TXnohead4
TXdesk1 $ whoami
install1

TXdesk1 $ ssh -X -l install1 TXnohead4
Password: Ins1PwD1
TXnohead4 $
```

To reach the global zone, the administrator assumes the role `remoterole`. This role is defined identically on both systems.

```
TXnohead4 # su - remoterole
Password: abcd1EFG
```

Then, the administrator starts the `txzonemgr` GUI.

```
TXnohead4 $ /usr/sbin/txzonemgr &
```

The Labeled Zone Manager runs on the headless system and displays on the desktop system.

PART II

Administration of Trusted Extensions

This section describes how to administer Trusted Extensions.

Chapter 7, "Trusted Extensions Administration Concepts," introduces you to Trusted Extensions software.

Chapter 8, "Trusted Extensions Administration Tools," describes the administrative programs that are specific to Trusted Extensions software.

Chapter 9, "Getting Started as a Trusted Extensions Administrator (Tasks)," introduces the administrator to Trusted Extensions. This chapter also describes the new features in this release.

Chapter 10, "Security Requirements on a Trusted Extensions System (Overview)," describes the .

Chapter 11, "Administering Security Requirements in Trusted Extensions (Tasks)," describes common tasks when administering Trusted Extensions.

Chapter 12, "Users, Rights, and Roles in Trusted Extensions (Overview)," introduces RBAC for Trusted Extensions.

Chapter 13, "Managing Users, Rights, and Roles in Trusted Extensions (Tasks)," provides instructions on managing regular users of Trusted Extensions.

Chapter 14, "Remote Administration in Trusted Extensions (Tasks)," provides instructions on administering Trusted Extensions remotely.

Chapter 15, "Trusted Extensions and LDAP (Overview)," provides instructions on administering LDAP for Trusted Extensions.

Chapter 16, "Managing Zones in Trusted Extensions (Tasks)," provides instructions on managing labeled zones.

Chapter 17, "Managing and Mounting Files in Trusted Extensions (Tasks)," provides instructions on managing mounting, backing up the system, and other file-related tasks in Trusted Extensions.

Chapter 18, "Trusted Networking (Overview)," provides an overview of the network databases and routing in Trusted Extensions.

Chapter 19, "Managing Networks in Trusted Extensions (Tasks)," provides instructions on managing the network databases and routing in Trusted Extensions.

Chapter 20, "Multilevel Mail in Trusted Extensions (Overview)," describes mail-specific issues in Trusted Extensions.

Chapter 21, "Managing Labeled Printing (Tasks)," provides instructions on handling printing in Trusted Extensions.

Chapter 23, "Managing Devices for Trusted Extensions (Tasks)," provides instructions on managing devices by using the Device Allocation Manager.

Chapter 24, "Trusted Extensions Auditing (Overview)," provides Trusted Extensions–specific information about auditing.

Chapter 25, "Software Management in Trusted Extensions (Tasks)," describes how to administer programs on a Trusted Extensions system.

CHAPTER 7

Trusted Extensions Administration Concepts

This chapter introduces you to administering a system that is configured with Solaris™ Trusted Extensions software.

Trusted Extensions Software and the Solaris OS

Trusted Extensions software adds labels to a system that is running the Solaris Operating System (Solaris OS). Labels implement *mandatory access control* (MAC). MAC, along with discretionary access control (DAC), protects system subjects (processes) and objects (data). Trusted Extensions software provides interfaces to handle label configuration, label assignment, and label policy.

Similarities Between Trusted Extensions and the Solaris OS

Trusted Extensions software uses rights profiles, roles, auditing, privileges, and other security features of the Solaris OS. You can use Solaris Secure Shell (SSH), BART, the Solaris cryptographic framework, IPsec, and IPfilter with Trusted Extensions.

- As in the Solaris OS, users can be limited to using applications that are necessary for performing their jobs. Other users can be authorized to do more.
- As in the Solaris OS, capabilities that were formerly assigned to superuser are assigned to separate, discrete "roles."
- As in the Solaris OS, privileges protect processes. Zones are also used to separate processes.
- As in the Solaris OS, events on the system can be audited.

- Trusted Extensions uses the system configuration files of the Solaris OS, such as `policy.conf` and `exec_attr`.

Differences Between Trusted Extensions and the Solaris OS

Trusted Extensions software extends the Solaris OS. The following list provides an overview. For a quick reference, see Appendix C, "Quick Reference to Trusted Extensions Administration."

- Trusted Extensions controls access to data with special security tags that are called *labels*. Labels provide *mandatory access control* (MAC). MAC protection is in addition to UNIX® file permissions, or discretionary access control (DAC). Labels are directly assigned to users, zones, devices, windows, and network endpoints. Labels are implicitly assigned to processes, files, and other system objects.

  MAC cannot be overridden by regular users. Trusted Extensions requires regular users to operate in labeled zones. By default, no users or processes in labeled zones can override MAC.

  As in the Solaris OS, the ability to override security policy can be assigned to specific processes or users when MAC can be overridden. For example, users can be authorized to change the label of a file. Such an action upgrades or downgrades the sensitivity of the information in that file.
- Trusted Extensions adds to existing configuration files and commands. For example, Trusted Extensions adds audit events, authorizations, privileges, and rights profiles.
- Some features that are optional on a Solaris system are required on a Trusted Extensions system. For example, zones and roles are required on a system that is configured with Trusted Extensions.
- Some features that are optional on a Solaris system are recommended on a Trusted Extensions system. For example, in Trusted Extensions the `root` user should be turned into the `root` role.
- Trusted Extensions can change the default behavior of the Solaris OS. For example, on a system that is configured with Trusted Extensions, auditing is enabled by default. In addition, device allocation is required.
- Trusted Extensions can narrow the options that are available in the Solaris OS. For example, on a system that is configured with Trusted Extensions, the NIS+ naming service is not supported. Also, in Trusted Extensions, all zones are labeled zones. Unlike the Solaris OS, labeled zones must use the same pool of user IDs and group IDs. Additionally, in Trusted Extensions, labeled zones can share one IP address.
- Trusted Extensions provides a trusted version of the GNOME desktop, Solaris Trusted Extensions (GNOME). The name can be shortened to Trusted GNOME.

- Trusted Extensions provides additional graphical user interfaces (GUIs) and command line interfaces (CLIs). For example, Trusted Extensions provides the Device Allocation Manager to administer devices. In addition, the `updatehome` command is used to place startup files in an regular user's home directory at every label.
- Trusted Extensions requires the use of particular GUIs for administration. For example, on a system that is configured with Trusted Extensions, the Solaris Management Console is used to administer users, roles, and the network.
- Trusted Extensions limits what users can see. For example, a device that cannot be allocated by a user cannot be seen by that user.
- Trusted Extensions limits users' desktop options. For example, users are allowed a limited time of workstation inactivity before the screen locks.

Multiheaded Systems and the Trusted Extensions Desktop

When the monitors of a multiheaded Trusted Extensions system are configured horizontally, the trusted stripe stretches across the monitors. When the monitors are configured vertically, the trusted stripe appears in the lowest monitor.

When different workspaces are displayed on the monitors of a multiheaded system, Trusted GNOME displays a trusted stripe on each monitor.

Basic Concepts of Trusted Extensions

Trusted Extensions software adds labels to a Solaris system. Labeled desktops and trusted applications, such as the Label Builder and the Device Allocation Manager, are also added. The concepts in this section are necessary to understand Trusted Extensions, both for users and administrators. Users are introduced to these concepts in the *Solaris Trusted Extensions User's Guide*.

Trusted Extensions Protections

Trusted Extensions software enhances the protection of the Solaris OS. The Solaris OS protects access to the system with user accounts that require passwords. You can require that passwords be changed regularly, be of a certain length, and so on. Roles require additional passwords to perform administrative tasks. Additional authentication limits the damage that can be done by an intruder who guesses the root password, because roles cannot be used as login accounts. Trusted Extensions software goes further by restricting users and roles to an approved label range. This label range limits the information that users and roles can access.

Trusted Extensions software displays the Trusted Path symbol, an unmistakable, tamper-proof emblem that appears at the left of the trusted stripe. In Trusted GNOME, the stripe is at the top of the screen. The Trusted Path symbol indicates to users when they are using security-related parts of the system. If this symbol does not appear when the user is running a trusted application, that version of the application should be checked immediately for authenticity. If the trusted stripe does not appear, the desktop is not trustworthy. For a sample desktop display, see Figure 7–1.

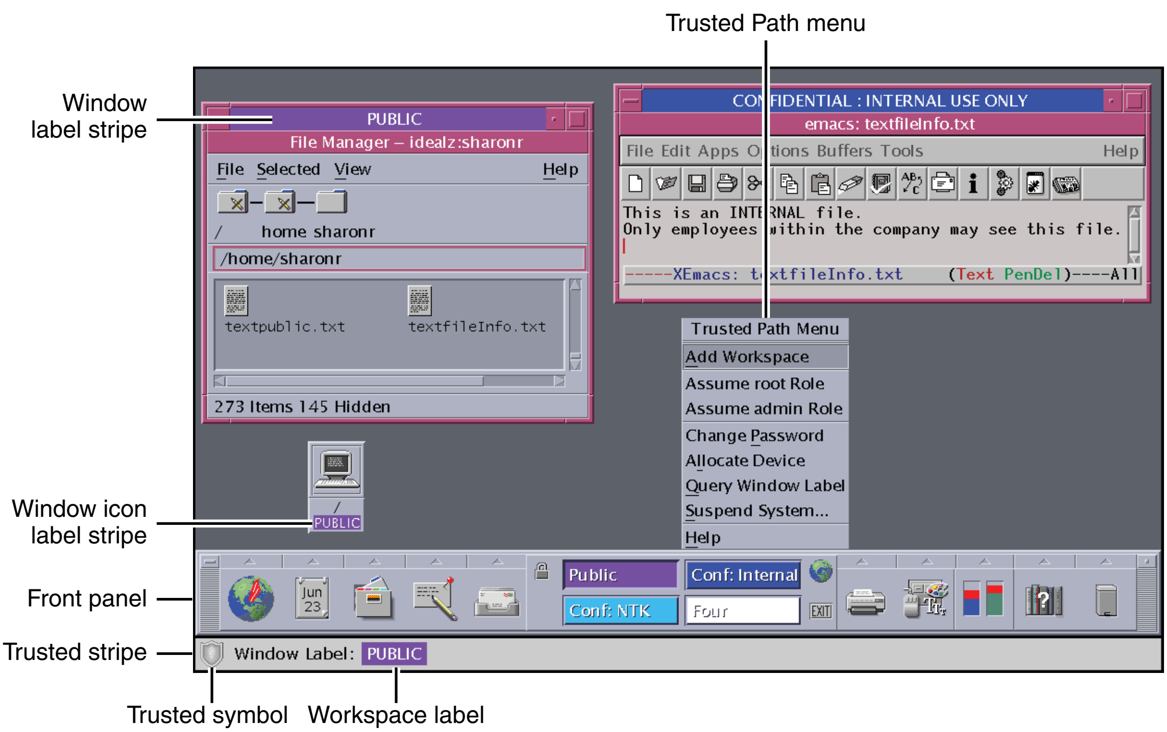


FIGURE 7–1 Trusted Extensions Multilevel CDE Desktop

Most security-related software, that is, the Trusted Computing Base (TCB), runs in the global zone. Regular users cannot enter the global zone or view its resources. Users are able to interact with TCB software, as in when they change passwords. The Trusted Path symbol is displayed whenever the user interacts with the TCB.

Trusted Extensions and Access Control

Trusted Extensions software protects information and other resources through both discretionary access control (DAC) and mandatory access control (MAC). DAC is the traditional UNIX permission bits and access control lists that are set at the discretion of the owner. MAC is a mechanism that the system enforces automatically. MAC controls all transactions by checking the labels of processes and data in the transaction.

A user's *label* represents the sensitivity level at which the user is permitted to operate and chooses to operate. Typical labels are `Secret`, or `Public`. The label determines the information that the user is allowed to access. Both MAC and DAC can be overridden by special permissions that are in the Solaris OS. *Privileges* are special permissions that can be granted to processes. *Authorizations* are special permissions that can be granted to users and roles by an administrator.

As an administrator, you need to train users on the proper procedures for securing their files and directories, according to your site's security policy. Furthermore, you need to instruct any users who are allowed to upgrade or downgrade labels as to when doing so is appropriate.

Roles and Trusted Extensions

On a system that is running Solaris software without Trusted Extensions, roles are optional. On a system that is configured with Trusted Extensions, roles are required. The system is administered by the System Administrator role and the Security Administrator role. In some cases, the `root` role is used.

As in the Solaris OS, rights profiles are the basis of a role's capabilities. Trusted Extensions provides two rights profiles, Information Security and User Security. These two profiles define the Security Administrator role.

The programs that are available to a role in Trusted Extensions have a special property, the *trusted path attribute*. This attribute indicates that the program is part of the TCB. The trusted path attribute is available when a program is launched from the global zone.

For information about roles, see Part III, "Roles, Rights Profiles, and Privileges," in *System Administration Guide: Security Services*.

Labels in Trusted Extensions Software

Labels and clearances are at the center of mandatory access control (MAC) in Trusted Extensions. They determine which users can access which programs, files, and directories. Labels and clearances consist of one *classification* component and zero or more *compartment* components. The classification component indicates a hierarchical level of security such as `TOP`

SECRET or CONFIDENTIAL. The compartment component represents a group of users who might need access to a common body of information. Some typical types of compartments are projects, departments, or physical locations. Labels are readable by authorized users, but internally, labels are manipulated as numbers. The numbers and their readable versions are defined in the label_encodings file.

Trusted Extensions mediates all attempted security-related transactions. The software compares the labels of the accessing entity, typically a process, and the entity being accessed, usually a filesystem object. The software then permits or disallows the transaction depending on which label is *dominant*. Labels are also used to determine access to other system resources, such as allocatable devices, networks, frame buffers, and other hosts.

Dominance Relationships Between Labels

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

- The classification component of the first entity's label is equal to or higher than the second entity's classification. The security administrator assigns numbers to classifications in the label_encodings file. The software compares these numbers to determine dominance.
- The set of compartments in the first entity includes all of the second entity's compartments.

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

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

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

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

TABLE 7–1 Examples of Label Relationships

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

TABLE 7–1 Examples of Label Relationships *(Continued)*

Label 1	Relationship	Label 2
`NEED_TO_KNOW Eng Mkt`	is disjoint with	`INTERNAL Eng Mkt Fin`

Administrative Labels

Trusted Extensions provides two special administrative labels that are used as labels or clearances: `ADMIN_HIGH` and `ADMIN_LOW`. These labels are used to protect system resources and are intended for administrators rather than regular users.

`ADMIN_HIGH` is the highest label. `ADMIN_HIGH` dominates all other labels in the system and is used to protect system data, such as administration databases or audit trails, from being read. You must be in the global zone to read data that is labeled `ADMIN_HIGH`.

`ADMIN_LOW` is the lowest label. `ADMIN_LOW` is dominated by all other labels in a system, including labels for regular users. Mandatory access control does not permit users to write data to files with labels lower than the user's label. Thus, a file at the label `ADMIN_LOW` can be read by regular users, but cannot be modified. `ADMIN_LOW` is typically used to protect public executables that are shared, such as files in `/usr/bin`.

Label Encodings File

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

- **Component definitions** – Definitions of classifications, compartments, labels, and clearances, including rules for required combinations and constraints
- **Accreditation range definitions** – Specification of the clearances and minimum labels that define the sets of available labels for the entire system and for regular users
- **Printing specifications** – Identification and handling information for print banners, trailers, headers, footers, and other security features on printer output
- **Customizations** – Local definitions including label color codes, and other defaults

For more information, see the `label_encodings(4)` man page. Detailed information can also be found in *Solaris Trusted Extensions Label Administration* and *Compartmented Mode Workstation Labeling: Encodings Format*.

Label Ranges

A *label range* is the set of potentially usable labels at which users can operate. Both users and resources both have label ranges. Resources that can be protected by label ranges include such things as allocatable devices, networks, interfaces, frame buffers, and commands. A label range is defined by a clearance at the top of the range and a minimum label at the bottom.

A range does not necessarily include all combinations of labels that fall between a maximum and minimum label. Rules in the `label_encodings` file can disqualify certain combinations. A label must be *well-formed*, that is, permitted by all applicable rules in the label encodings file, in order to be included in a range.

However, a clearance does not have to be well-formed. Suppose, for example, that a `label_encodings` file prohibits any combination of compartments `Eng`, `Mkt`, and `Fin` in a label. `INTERNAL Eng Mkt Fin` would be a valid clearance but not a valid label. As a clearance, this combination would let a user access files that are labeled `INTERNAL Eng`, `INTERNAL Mkt`, and `INTERNAL Fin`.

Account Label Range

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

minimum label ≤ permitted label ≤ clearance

Thus, the user is permitted to operate at any label that is dominated by the clearance as long as that label dominates the minimum label. When a user's clearance or minimum label is not expressly set, the defaults that are defined in the `label_encodings` file take effect.

Users can be assigned a clearance and a minimum label that enable them to operate at more than one label, or at a single label. When a user's clearance and minimum label are equal, the user can operate at only one label.

Session Range

The *session range* is the set of labels that is available to a user during a Trusted Extensions session. The session range must be within the user's account label range and the label range set for the system. At login, if the user selects single-label session mode, the session range is limited to that label. If the user selects multilabel session mode, then the label that the user selects becomes the session clearance. The session clearance defines the upper boundary of the session range. The user's minimum label defines the lower bound. The user begins the session in a workspace at the minimum label. During the session, the user can switch to a workspace at any label within the session range.

What Labels Protect and Where Labels Appear

Labels appear on the desktop and on output that is executed on the desktop, such as printer output.

- **Applications –** Applications start processes. These processes run at the label of the workspace where the application is started. An application in a labeled zone, as a file, is labeled at the label of the zone.

- **Devices –** Data flowing through devices is controlled through device allocation and device label ranges. To use a device, users must be within the label range of the device, and be authorized to allocate the device.
- **File system mount points –** Every mount point has a label. The label is viewable by using the `getlabel` command.
- **IPsec and IKE –** IPsec security associations and IKE rules have labels.
- **Network interfaces –** IP addresses (hosts) have templates that describe their label range. Unlabeled hosts also have a default label.
- **Printers and printing –** Printers have label ranges. Labels are printed on body pages. Labels, handling information, and other security information is printed on the banner and trailer pages. To configure printing in Trusted Extensions, see Chapter 21, "Managing Labeled Printing (Tasks)," and "Labels on Printed Output" in *Solaris Trusted Extensions Label Administration*.
- **Processes –** Processes are labeled. Processes run at the label of the workspace where the process originates. The label of a process is visible by using the `plabel` command.
- **Users –** Users are assigned a default label and a label range. The label of the user's workspace indicates the label of the user's processes.
- **Windows –** Labels are visible at the top of desktop windows. The label of the desktop is also indicated by color. The color appears on the desktop switch and above window title bars.

  When a window is moved to a differently labeled workspace, the window maintains its original label.
- **Zones –** Every zone has a unique label. The files and directories that are owned by a zone are at the zone's label. For more information, see the `getzonepath(1)` man page.

CHAPTER 8

Trusted Extensions Administration Tools

This chapter describes the tools that are available in Solaris Trusted Extensions, the location of the tools, and the databases on which the tools operate.

Administration Tools for Trusted Extensions

Administration on a system that is configured with Trusted Extensions uses many of the same tools that are available in the Solaris OS. Trusted Extensions offers security-enhanced tools as well. Administration tools are available only to roles in a role workspace.

Within a role workspace, you can access commands, applications, and scripts that are trusted. The following table summarizes these administrative tools.

TABLE 8–1 Trusted Extensions Administrative Tools

Tool	Description	For More Information
`/usr/sbin/txzonemgr`	Provides a menu-based wizard for creating, installing, initializing, and booting zones. The script also provides menu items for networking options, name services options, and for clienting the global zone to an existing LDAP server. `txzonemgr` uses the `zenity` command.	See "Creating Labeled Zones" on page 65 See also the `zenity`(1) man page.

TABLE 8–1 Trusted Extensions Administrative Tools *(Continued)*

Tool	Description	For More Information
In Solaris Trusted Extensions (GNOME), Device Manager	Used to administer the label ranges of devices, and to allocate or deallocate devices.	See "Device Manager" on page 140 and "Handling Devices in Trusted Extensions (Task Map)" on page 329.
Solaris Management Console	Used to configure users, roles, rights, hosts, zones, and networks. This tool can update local files or LDAP databases. This tool can also launch the `dtappsession` legacy application.	For basic functionality, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*. For information that is specific to Trusted Extensions, see "Solaris Management Console Tools" on page 141.
Solaris Management Console commands, such as `smuser` and `smtnzonecfg`	Is the command-line interface for the Solaris Management Console.	For a list, see Table 8–2.
Label Builder	Is also a user tool. Appears when a program requires you to choose a label.	For an example, see "How to Modify a User's Label Range in the Solaris Management Console" on page 189.
Trusted Extensions commands	Used to perform tasks that are not covered by Solaris Management Console tools.	For the list of administrative commands, see Table 8–3.

`txzonemgr` Script

In the Solaris Express Community Edition, the `txzonemgr` script is used to configure labeled zones. This `zenity(1)` script displays a dialog box with the title Labeled Zone Manager. This GUI presents a dynamically-determined menu that displays only valid choices for the current configuration status of a labeled zone. For instance, if a zone is already labeled, the Label menu item is not displayed.

Device Manager

A *device* is either a physical peripheral that is connected to a computer or a software-simulated device called a *pseudo-device*. Because devices provide a means for the import and export of data to and from a system, devices must be controlled to properly protect the data. Trusted Extensions uses device allocation and device label ranges to control data flowing through devices.

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

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

You can manage devices by using the Device Administration tool from the Device Manager. Regular users cannot access the Device Administration tool.

For more information about device protection in Trusted Extensions, see Chapter 23, "Managing Devices for Trusted Extensions (Tasks)."

Solaris Management Console Tools

The Solaris Management Console provides access to toolboxes of GUI-based administration tools. These tools enable you to edit items in various configuration databases. In Trusted Extensions, the Solaris Management Console is the administrative interface for users, roles, and the trusted network databases.

Trusted Extensions extends the Solaris Management Console:

- Trusted Extensions modifies the Solaris Management Console Users tool set. For an introduction to the tool set, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.
- Trusted Extensions adds the Security Templates tool and the Trusted Network Zones tool to the Computers and Networks tool set.

Solaris Management Console tools are collected into *toolboxes* according to scope and security policy. To administer Trusted Extensions, Trusted Extensions provides toolboxes whose `Policy=TSOL`. You can access tools according to scope, that is, according to naming service. The available scopes are local host and LDAP.

The Solaris Management Console is shown in the following figure. A `Scope=Files` Trusted Extensions toolbox is loaded, and the Users tool set is open.

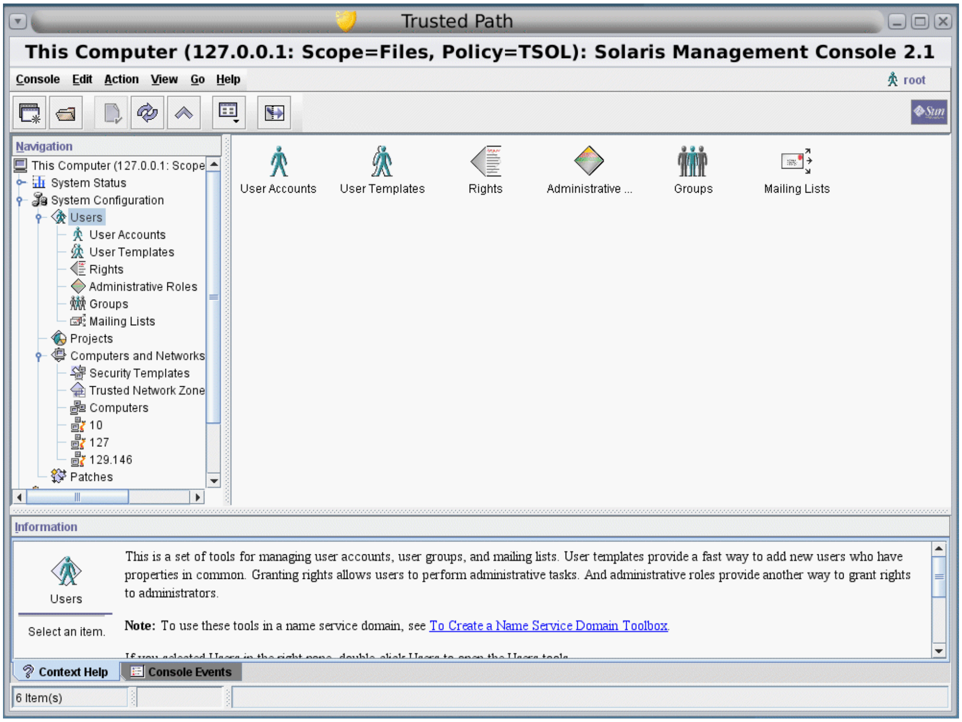


FIGURE 8–1 Typical Trusted Extensions Toolbox in the Solaris Management Console

Trusted Extensions Tools in the Solaris Management Console

Trusted Extensions adds configurable security attributes to three tools:

- **User Accounts tool** – Is the administrative interface to change a user's label, change a user's view of labels, and to control account usage.
- **Administrative Roles tool** – Is the administrative interface to change a role's label range and screen-locking behavior when idle.

Trusted Extensions adds two tools to the Computers and Networks tool set:

- **Security Templates tool** – Is the administrative interface for managing the label aspects of hosts and networks. This tool modifies the `tnrhtp` and `tnrhdb` databases, enforces syntactic accuracy, and updates the kernel with the changes.
- **Trusted Network Zones tool** – Is the administrative interface for managing the label aspects of zones. This tool modifies the `tnzonecfg` database, enforces syntactic accuracy, and updates the kernel with the changes.

Figure 8–2 shows the Files toolbox with the Users tool set highlighted. The Trusted Extensions tools appear below the Computers and Networks tool set.

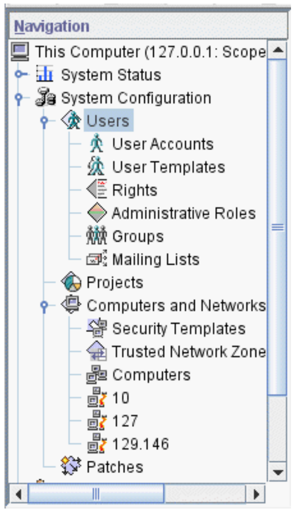


FIGURE 8–2 Computers and Networks Tool Set in the Solaris Management Console

Security Templates Tool

A *security template* describes a set of security attributes that can be assigned to a group of hosts. The Security Templates tool enables you to conveniently assign a specific combination of security attributes to a group of hosts. These attributes control how data is packaged, transmitted, and interpreted. Hosts that are assigned to a template have identical security settings.

The hosts are defined in the Computers tool. The security attributes of the hosts are assigned in the Security Templates tool. The Modify Template dialog box contains two tabs:

- **General tab –** Describes the template. Includes its name, host type, default label, domain of interpretation (DOI), accreditation range, and set of discrete sensitivity labels.
- **Hosts Assigned to Template tab –** Lists all the hosts on the network that you have assigned to this template.

Trusted networking and security templates are explained in more detail in Chapter 18, "Trusted Networking (Overview)."

Trusted Network Zones Tool

The Trusted Network Zones tool identifies the zones on your system. Initially, the global zone is listed. When you add zones and their labels, the zone names display in the pane. Zone creation usually occurs during system configuration. Label assignment, multilevel port configuration, and label policy is configured in this tool. For details, see Chapter 16, "Managing Zones in Trusted Extensions (Tasks)."

Client-Server Communication With the Solaris Management Console

Typically, a Solaris Management Console client administers systems *remotely*. On a network that uses LDAP as a naming service, a Solaris Management Console client connects to the Solaris Management Console server that runs on the LDAP server. The following figure shows this configuration.

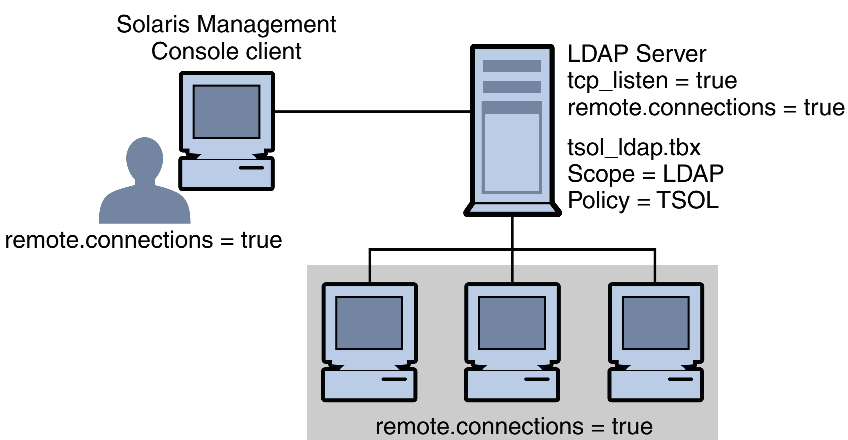


FIGURE 8–3 Solaris Management Console Client Using an LDAP Server to Administer the Network

Figure 8–4 shows a network that is not configured with an LDAP server. The administrator configured each remote system with a Solaris Management Console server.

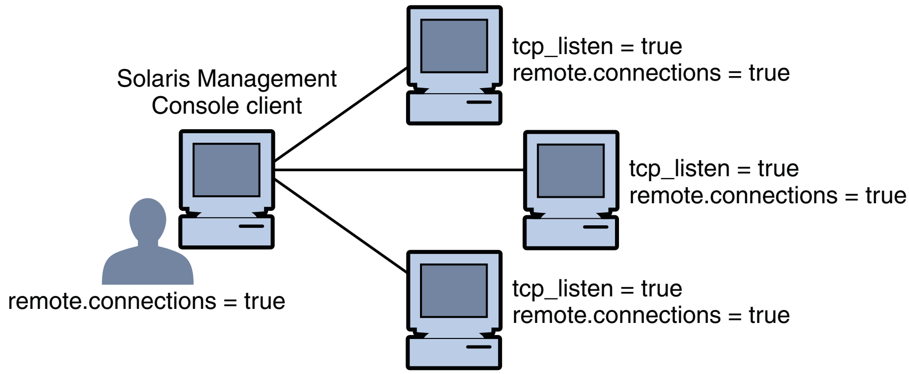


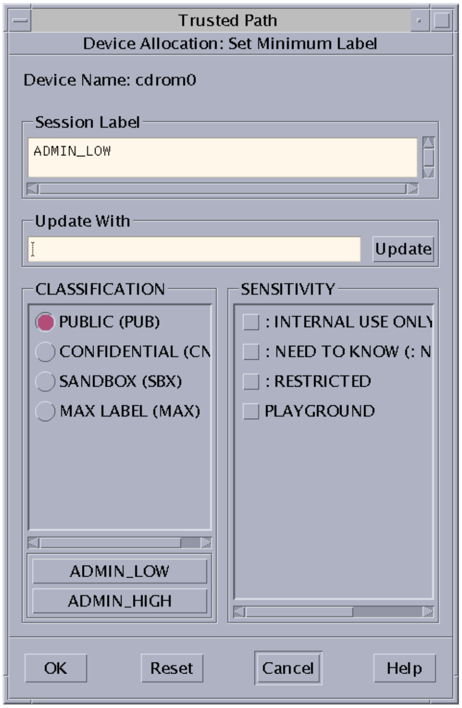
FIGURE 8–4 Solaris Management Console Client Administering Individual Remote Systems on a Network

Solaris Management Console Documentation

The main source of documentation for the Solaris Management Console is its online help. Context-sensitive help is tied to the currently selected feature and is displayed in the information pane. Expanded help topics are available from the Help menu or by clicking links in the context-sensitive help. Further information is provided in Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*. Also see "Using the Solaris Management Tools With RBAC (Task Map)" in *System Administration Guide: Basic Administration*.

Label Builder in Trusted Extensions

The label builder GUI enforces your choice of a valid label or clearance when a program requires you to assign a label. For example, a label builder appears during login (see Chapter 3, "Logging In to Trusted Extensions (Tasks)," in *Solaris Trusted Extensions User's Guide*). The label builder also appears when you change the label of a workspace, or when you assign a label to a user, zone, or network interface in the Solaris Management Console. The following label builder appears when you assign a label range to a new device.



In the label builder, component names in the Classification column correspond to the `CLASSIFICATIONS` section in the `label_encodings` file. The component names in the Sensitivity column correspond to the `WORDS` section in the `label_encodings` file.

Command Line Tools in Trusted Extensions

Commands that are unique to Trusted Extensions are contained in the *Solaris Trusted Extensions Reference Manual*. The Solaris commands that Trusted Extensions modifies are contained in the *Solaris Reference Manual*. The `man` command finds all the commands.

The following table lists commands that are unique to Trusted Extensions. The commands are listed in man page format.

TABLE 8–2 User and Administrative Trusted Extensions Commands

Man Page	Trusted Extensions Modification	For More Information
`add_allocatable(1M)`	Enables a device to be allocated by adding the device to device allocation databases. By default, removable devices are allocatable.	"How to Configure a Device in Trusted Extensions" on page 331

TABLE 8–2 User and Administrative Trusted Extensions Commands *(Continued)*

Man Page	Trusted Extensions Modification	For More Information
`atohexlabel(1M)`	Translates a label into hexadecimal format.	"How to Obtain the Hexadecimal Equivalent for a Label" on page 168
`chk_encodings(1M)`	Checks the integrity of the `label_encodings` file.	"How to Debug a label_encodings File" in *Solaris Trusted Extensions Label Administration*
`getlabel(1)`	Displays the label of the selected files or directories.	"How to Display the Labels of Mounted Files" on page 220
`getzonepath(1)`	Displays the full pathname of a specific zone.	"Acquiring a Sensitivity Label" in *Solaris Trusted Extensions Developer's Guide*
`hextoalabel(1M)`	Translates a hexadecimal label into its readable equivalent.	"How to Obtain a Readable Label From Its Hexadecimal Form" on page 170
`plabel(1)`	Displays the label of the current process.	See the man page.
`remove_allocatable(1M)`	Prevents allocation of a device by removing its entry from device allocation databases.	"How to Configure a Device in Trusted Extensions" on page 331
`setlabel(1)`	Relabels the selected item. Requires the `solaris.label.file.downgrade` or `solaris.label.file.upgrade` authorization. These authorizations are in the Object Label Management rights profile.	
`smtnrhdb(1M)`	Manages entries in the `tnrhdb` database locally or in a naming service database.	For equivalent procedures that use the Solaris Management Console, see "Configuring Trusted Network Databases (Task Map)" on page 270.
`smtnrhtp(1M)`	Manages entries in the `tnrhtp` database locally or in a naming service database.	See the man page.
`smtnzonecfg(1M)`	Manages entries in the local `tnzonecfg` database.	For an equivalent procedure that uses the Solaris Management Console, see "How to Create a Multilevel Port for a Zone" on page 228.
`tnchkdb(1M)`	Checks the integrity of the `tnrhdb` and `tnrhtp` databases.	"How to Check the Syntax of Trusted Network Databases" on page 285
`tnctl(1M)`	Caches network information in the kernel.	"How to Synchronize the Kernel Cache With Trusted Network Databases" on page 287

TABLE 8–2 User and Administrative Trusted Extensions Commands *(Continued)*

Man Page	Trusted Extensions Modification	For More Information
tnd(1M)	Executes the trusted network daemon.	"How to Synchronize the Kernel Cache With Trusted Network Databases" on page 287
tninfo(1M)	Displays kernel-level network information and statistics.	"How to Compare Trusted Network Database Information With the Kernel Cache" on page 286.
updatehome(1M)	Updates `.copy_files` and `.link_files` for the current label.	"How to Configure Startup Files for Users in Trusted Extensions" on page 184

The following table lists Solaris commands that are modified or extended by Trusted Extensions. The commands are listed in man page format.

TABLE 8–3 User and Administrative Commands That Trusted Extensions Modifies

Man Page	Purpose of Command	For More Information
allocate(1)	Adds options to clean the allocated device, and to allocate a device to a specific zone. In Trusted Extensions, regular users do not use this command.	"How to Allocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*
deallocate(1)	Adds options to clean the device, and to deallocate a device from a specific zone. In Trusted Extensions, regular users do not use this command.	"How to Allocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*
list_devices(1)	Adds the `-a` option to display device attributes, such as authorizations and labels. Adds the `-d` option to display the default attributes of an allocated device type. Adds the `-z` option to display available devices that can be allocated to a labeled zone.	See the man page.
tar(1)	Adds the `-T` option to archive and extract files and directories that are labeled.	"How to Back Up Files in Trusted Extensions" on page 238 and "How to Restore Files in Trusted Extensions" on page 239
auditconfig(1M)	Adds the `windata_down` and `windata_up` audit policy options.	"How to Configure Audit Policy" in *System Administration Guide: Security Services*
auditreduce(1M)	Adds the `-l` option to select audit records by label.	"How to Select Audit Events From the Audit Trail" in *System Administration Guide: Security Services*
automount(1M)	Modifies the names and contents of `auto_home` maps to account for zone names and zone visibility from higher labels.	"Changes to the Automounter in Trusted Extensions" on page 235

TABLE 8–3 User and Administrative Commands That Trusted Extensions Modifies *(Continued)*

Man Page	Purpose of Command	For More Information
ifconfig(1M)	Adds the all-zones option to make an interface available to every zone on the system.	"How to Verify That a Host's Interfaces Are Up" on page 294
netstat(1M)	Adds the -R option to display extended security attributes for sockets and routing table entries.	"How to Debug the Trusted Extensions Network" on page 295
route(1M)	Adds the -secattr option to display the security attributes of the route: cipso, doi, max_sl, and min_sl.	"How to Configure Routes With Security Attributes" on page 284
ikeadm(1M)	Adds a debug flag, 0x0400, for label processing.	See the man page.
in.iked(1M)	In the global zone, uses two multilevel ports, UDP ports 500 and 4500, to negotiate labeled security associations.	See the ike.config(4) man page.
ipseckey(1M)	Adds the label, outer-label, and implicit-label extensions. These extensions associate Trusted Extensions labels with the traffic that is carried inside a security association.	See the man page.

Configuration Files in Trusted Extensions

The following Solaris configuration files are modified or extended by Trusted Extensions. The files are introduced in man page format.

- ike.config(4) – Trusted Extensions adds the label_aware global parameter and three Phase 1 transform parameters, single_label and multi_label, and wire_label.

Note – The IKE configuration file contains a keyword, label, that is used to make a Phase 1 IKE rule unique. The IKE keyword label is distinct from Trusted Extensions labels.

Remote Administration in Trusted Extensions

You can remotely administer a system that is configured with Trusted Extensions by using the ssh command, the dtappsession program, or the Solaris Management Console. If site security policy permits, you can configure a Trusted Extensions host to enable login from a non-Trusted Extensions host, although this configuration is less secure. For more information, see Chapter 14, "Remote Administration in Trusted Extensions (Tasks)."

◆ ◆ ◆ CHAPTER 9

Getting Started as a Trusted Extensions Administrator (Tasks)

This chapter introduces you to administering a system that is configured with Solaris Trusted Extensions.

- "Security Requirements When Administering Trusted Extensions" on page 151
- "Getting Started as a Trusted Extensions Administrator (Task Map)" on page 152

Security Requirements When Administering Trusted Extensions

In Trusted Extensions, roles are the conventional way to administer the system. Typically, superuser is not used. Roles are created just as they are in the Solaris OS, and most tasks are performed by roles. In Trusted Extensions, the `root` user is not used to perform administrative tasks.

The following roles are typical of a Trusted Extensions site:

- `root` **role** – Created by the initial setup team
- **Security Administrator role** – Created during or after initial configuration by the initial setup team
- **System Administrator role** – Created by the Security Administrator role

As in the Solaris OS, you might also create a Primary Administrator role, an Operator role, and so on. With the exception of the `root` role, the roles that you create can be administered in a naming service.

As in the Solaris OS, only users who have been assigned a role can assume that role. In Solaris Trusted Extensions (GNOME), you can assume a role when your user name is displayed in the Trusted Stripe. The role choices appear when you click your user name.

Role Creation in Trusted Extensions

To administer Trusted Extensions, you create roles that divide system and security functions. The initial setup team created the Security Administrator role during configuration. For details, see "Create the Security Administrator Role in Trusted Extensions" on page 83.

The process of creating a role in Trusted Extensions is identical to the Solaris OS process. As described in Chapter 8, "Trusted Extensions Administration Tools," the Solaris Management Console is the GUI for managing roles in Trusted Extensions.

- For an overview of role creation, see Chapter 10, "Role-Based Access Control (Reference)," in *System Administration Guide: Security Services* and "Using RBAC (Task Map)" in *System Administration Guide: Security Services*.
- To create a powerful role that is equivalent to superuser, see "Creating the Primary Administrator Role" in *System Administration Guide: Basic Administration*. At sites that use Trusted Extensions, the Primary Administrator role might violate security policy. These sites would turn `root` into a role, and create a Security Administrator role.
- To create the `root` role, see "How to Make root User Into a Role" in *System Administration Guide: Security Services*.
- To create roles by using the Solaris Management Console, see "How to Create and Assign a Role by Using the GUI" in *System Administration Guide: Security Services*.

Role Assumption in Trusted Extensions

Unlike the Solaris OS, Trusted Extensions provides an Assume *Rolename* Role menu item from the Trusted Path menu. After confirming the role password, the software activates a role workspace with the trusted path attribute. Role workspaces are administrative workspaces. Such workspaces are in the global zone.

Getting Started as a Trusted Extensions Administrator (Task Map)

Familiarize yourself with the following procedures before administering Trusted Extensions.

Task	Description	For Instructions
Log in.	Logs you in securely.	"Logging In to Trusted Extensions" in *Solaris Trusted Extensions User's Guide*

Task	Description	For Instructions
Perform common user tasks on a desktop.	These tasks include: ■ Configuring your workspaces ■ Using workspaces at different labels ■ Accessing Trusted Extensions man pages ■ Accessing Trusted Extensions online help	"Working on a Labeled System" in *Solaris Trusted Extensions User's Guide*
Perform tasks that require the trusted path.	These tasks include: ■ Allocating a device ■ Changing your password ■ Changing the label of a workspace	"Performing Trusted Actions" in *Solaris Trusted Extensions User's Guide*
Create useful roles.	Creates administrative roles for your site. Creating roles in LDAP is a one-time task.	"Role Creation in Trusted Extensions" on page 152
	The Security Administrator role is a useful role.	"Create the Security Administrator Role in Trusted Extensions" on page 83
(Optional) Make `root` a role.	Prevents anonymous login by `root`. This task is done once per system.	"How to Make root User Into a Role" in *System Administration Guide: Security Services*
Assume a role.	Enters the global zone in a role. All administrative tasks are performed in the global zone.	"How to Enter the Global Zone in Trusted Extensions" on page 153
Exit a role workspace and become regular user.	Leaves the global zone.	"How to Exit the Global Zone in Trusted Extensions" on page 154
Locally administer users, roles, rights, zones, and networks.	Uses the Solaris Management Console to manage the distributed system.	"How to Administer the Local System With the Solaris Management Console" on page 154
Edit an administrative file.	Edits files in a trusted editor.	"How to Edit Administrative Files in Trusted Extensions" on page 155
Administer device allocation.	Uses the Device Manager – Administration GUI.	"Managing Devices in Trusted Extensions (Task Map)" on page 330

▼ How to Enter the Global Zone in Trusted Extensions

By assuming a role, you enter the global zone in Trusted Extensions. Administration of the entire system is possible only from the global zone. Only superuser or a role can enter the global zone.

After assuming a role, the role can create a workspace at a user label to edit administration files in a labeled zone.

For troubleshooting purposes, you can also enter the global zone by starting a Failsafe session. For details, see "How to Log In to a Failsafe Session in Trusted Extensions" on page 188.

Before You Begin You have created one or more roles, or you plan to enter the global zone as superuser. For pointers, see "Role Creation in Trusted Extensions" on page 152.

1. **Use a trusted mechanism.**

   In Solaris Trusted Extensions (GNOME), click your user name in the trusted stripe and choose a role.

   If you have been assigned a role, the role names are displayed in a list.

   For the location and significance of Trusted Extensions desktop features, see Chapter 5, "Elements of Trusted Extensions (Reference)," in *Solaris Trusted Extensions User's Guide*.

2. **At the prompt, type the role password.**

   In Trusted GNOME, the current workspace changes to the role workspace.

   Click the role name on the trusted stripe, and from the menu, select a different role or user. This action changes the current workspace to the process of the new role or user.

▼ How to Exit the Global Zone in Trusted Extensions

Before You Begin You are in the global zone.

- **On the Trusted GNOME desktop, click your role name in the trusted stripe.**

  You can also exit the role workspace from the role menu.

  When you click the role name, your user name and a list of roles that you can assume is displayed. When you select your user name, all subsequent windows that you create in that workspace are created by the selected name. The windows that you previously created on the current desktop continue to display at the name and label of the role.

▼ How to Administer the Local System With the Solaris Management Console

The first time that you launch the Solaris Management Console on a system, a delay occurs while the tools are registered and various directories are created. This delay typically occurs during system configuration. For the procedure, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

To administer a remote system, see "Administering Trusted Extensions Remotely (Task Map)" on page 202.

Before You Begin You must have assumed a role. For details, see "How to Enter the Global Zone in Trusted Extensions" on page 153.

1. **Start the Solaris Management Console.**

   In Solaris Trusted Extensions (GNOME), use the command line.

   ```
   $ /usr/sbin/smc &
   ```

2. **Choose Console -> Open Toolbox.**

3. **From the list, select a Trusted Extensions toolbox of the appropriate scope.**

   A Trusted Extensions toolbox has `Policy=TSOL` as part of its name. The Files scope updates local files on the current system. The LDAP scope updates LDAP directories on the Sun Java™ System Directory Server. The toolbox names appear similar to the following:

   ```
   This Computer (this-host: Scope=Files, Policy=TSOL)
   This Computer (ldap-server: Scope=LDAP, Policy=TSOL)
   ```

4. **Navigate to the desired Solaris Management Console tool.**

   The password prompt is displayed.

   For tools that Trusted Extensions has modified, click System Configuration.

5. **Type the password.**

   Refer to the online help for additional information about Solaris Management Console tools. For an introduction to the tools that Trusted Extensions modifies, see "Solaris Management Console Tools" on page 141.

6. **To close the GUI, choose Exit from the Console menu.**

▼ How to Edit Administrative Files in Trusted Extensions

Administrative files are edited with a trusted editor that incorporates auditing. This editor also prevents the user from executing shell commands and from saving to any file name other than the name of the original file.

1. **Assume a role.**

   For details, see "How to Enter the Global Zone in Trusted Extensions" on page 153.

2 Open a trusted editor.

In Solaris Trusted Extensions (GNOME), do the following:

- **(Optional) To use `gedit` as the trusted editor, modify the `EDITOR` variable.**

  For details, see "How to Assign the Editor of Your Choice as the Trusted Editor" on page 166.

- **Use the command line to bring up the trusted editor.**

  ```
  # /usr/dt/bin/trusted_edit filename
  ```

  You must provide a *filename* argument.

3 To create a new file, type the full path name for the new file.

When you save the file, the editor creates a temporary file.

4 To edit an existing file, type the full path name for the existing file.

Note – If your editor provides a Save As option, do not use it. Use the editor's Save option to save the file.

---

5 To save the file to the specified path name, close the editor.

CHAPTER 10

Security Requirements on a Trusted Extensions System (Overview)

This chapter describes configurable security features on a system that is configured with Solaris Trusted Extensions.

Configurable Solaris Security Features

Trusted Extensions uses the same security features that the Solaris OS provides, and adds some features. For example, the Solaris OS provides `eeprom` protection, password requirements and strong password algorithms, system protection by locking out a user, and protection from keyboard shutdown.

Trusted Extensions differs from the Solaris OS in the actual procedures that are used to modify these security defaults. In Trusted Extensions, you typically administer systems by assuming a role. Local settings are modified by using the trusted editor. Changes that affect the network of users, roles, and hosts are made in the Solaris Management Console.

Trusted Extensions Interfaces for Configuring Security Features

Procedures are provided in this book where Trusted Extensions requires a particular interface to modify security settings, and that interface is optional in the Solaris OS. Where Trusted Extensions requires the use of the trusted editor to edit local files, no separate procedures are provided in this book. For example, the procedure "How to Prevent Account Locking for Users" on page 194 describes how to update a user's account by using the Solaris Management Console to prevent the account from being locked. However, the procedure for setting a system-wide password lock policy is not provided in this book. You follow the Solaris instructions, except that in Trusted Extensions, you use the trusted editor to modify the system file.

Extension of Solaris Security Mechanisms by Trusted Extensions

The following Solaris security mechanisms are extensible in Trusted Extensions as they are in the Solaris OS:

- **Audit events and classes** – Adding audit events and audit classes is described in Chapter 30, "Managing Solaris Auditing (Tasks)," in *System Administration Guide: Security Services*.
- **Rights profiles** – Adding rights profiles is described in Part III, "Roles, Rights Profiles, and Privileges," in *System Administration Guide: Security Services*.
- **Roles** – Adding roles is described in Part III, "Roles, Rights Profiles, and Privileges," in *System Administration Guide: Security Services*.
- **Authorizations** – For an example of adding a new authorization, see "Customizing Device Authorizations in Trusted Extensions (Task Map)" on page 337.

As in the Solaris OS, privileges cannot be extended.

Trusted Extensions Security Features

Trusted Extensions provides the following unique security features:

- **Labels** – Subjects and objects are labeled. Processes are labeled. Zones and the network are labeled.
- **Device Allocation Manager** – By default, devices are protected by allocation requirements. The Device Allocation Manager GUI is the interface for administrators and for regular users.
- **Change Password menu item** – The Trusted Path menu enables you to change your user password, and the password of the role that you have assumed.

Security Requirements Enforcement

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

Users and Security Requirements

Each site's security administrator ensures that users are trained in security procedures. The security administrator needs to communicate the following rules to new employees and remind existing employees of these rules on a regular basis:

- Do not tell anyone your password.

  Anyone who knows your password can access the same information that you can without being identified and therefore without being accountable.
- Do not write your password down or include it in an email message.
- Choose passwords that are hard to guess.
- Do not send your password to anyone by email.
- Do not leave your computer unattended without locking the screen or logging off.
- Remember that administrators do not rely on email to send instructions to users. Do not ever follow emailed instructions from an administrator without first double-checking with the administrator.

  Be aware that sender information in email can be forged.
- Because you are responsible for the access permissions on files and directories that you create, make sure that the permissions on your files and directories are set appropriately. Do not allow unauthorized users to read a file, to change a file, to list the contents of a directory, or to add to a directory.

Your site might want to provide additional suggestions.

Email Usage

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

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

Password Enforcement

The System Administrator role must specify a unique user name and user ID when creating a new account. When choosing the name and ID for a new account, the administrator you must ensure that both the user name and associated ID are not duplicated anywhere on the network and have not been previously used.

The Security Administrator role is responsible for specifying the original password for each account and for communicating the passwords to users of new accounts. You must consider the following information when administering passwords:

- Make sure that the accounts for users who are able to assume the Security Administrator role are configured so that the account cannot be locked. This practice ensures that at least one account can always log in and assume the Security Administrator role to reopen everyone's account if all other accounts are locked.
- Communicate the password to the user of a new account in such a way that the password cannot be eavesdropped by anyone else.
- Change an account's password if you have any suspicion that the password has been discovered by someone who should not know it.
- Never reuse user names or user IDs over the lifetime of the system.

  Ensuring that user names and user IDs are not reused prevents possible confusion about the following:

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

Information Protection

You as an administrator are responsible for correctly setting up and maintaining discretionary access control (DAC) and mandatory access control (MAC) protections for security-critical files. Critical files include the following:

- `shadow` **file** – Contains encrypted passwords. See `shadow(4)`.
- `prof_attr` **database** – Contains definitions of rights profiles. See `prof_attr(4)`.
- `exec_attr` **database** – Contains commands that are part of rights profiles. See `exec_attr(4)`.
- `user_attr` **file** – Contains the rights profiles, privileges, and authorizations that are assigned to local users. See `user_attr(4)`.
- **Audit trail** – Contains the audit records that the auditing service has collected. See `audit.log(4)`

Caution – Because the protection mechanisms for LDAP entries are not subject to the access control policy enforced by the Trusted Extensions software, the default LDAP entries must not be extended, and their access rules must not be modified.

Password Protection

In local files, passwords are protected from viewing by DAC and from modifications by both DAC and MAC. Passwords for local accounts are maintained in the `/etc/shadow` file, which is readable only by superuser. For more information, see the `shadow(4)` man page.

Group Administration

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

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

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

User Deletion Practices

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

- Delete the account's home directories in every zone.
- Delete any processes or jobs that are owned by the deleted account:
  - Delete any objects that are owned by the account,or assign the ownership to another user.
  - Delete any `at` or `batch` jobs that are scheduled on behalf of the user. For details, see the `at(1)` and `crontab(1)` man pages.
- Never reuse the user (account) name or user ID.

Rules When Changing the Level of Security for Data

By default, regular users can perform cut-and-paste, copy-and-paste, and drag-and-drop operations on both files and selections. The source and target must be at the same label.

To change the label of files, or the label of information within files requires authorization. When users are authorized to change the security level of data, the Selection Manager application mediates the transfer. The `/usr/share/gnome/sel_config` file controls file relabeling actions, and the cutting and copying of information to a different label. The `/usr/bin/tsoljdsselmgr` application controls drag-and-drop operations between windows. As the following tables illustrate, the relabeling of a selection is more restrictive than the relabeling of a file.

The following table summarizes the rules for file relabeling. The rules cover cut-and-paste, copy-and-paste, and drag-and-drop operations.

TABLE 10–1 Conditions for Moving Files to a New Label

Transaction Description	Label Relationship	Owner Relationship	Required Authorization
Copy and paste, cut and paste, or drag and drop of files between File Managers	Same label	Same UID	None
	Downgrade	Same UID	`solaris.label.file.downgrade`
	Upgrade	Same UID	`solaris.label.file.upgrade`
	Downgrade	Different UIDs	`solaris.label.file.downgrade`
	Upgrade	Different UIDs	`solaris.label.file.upgrade`

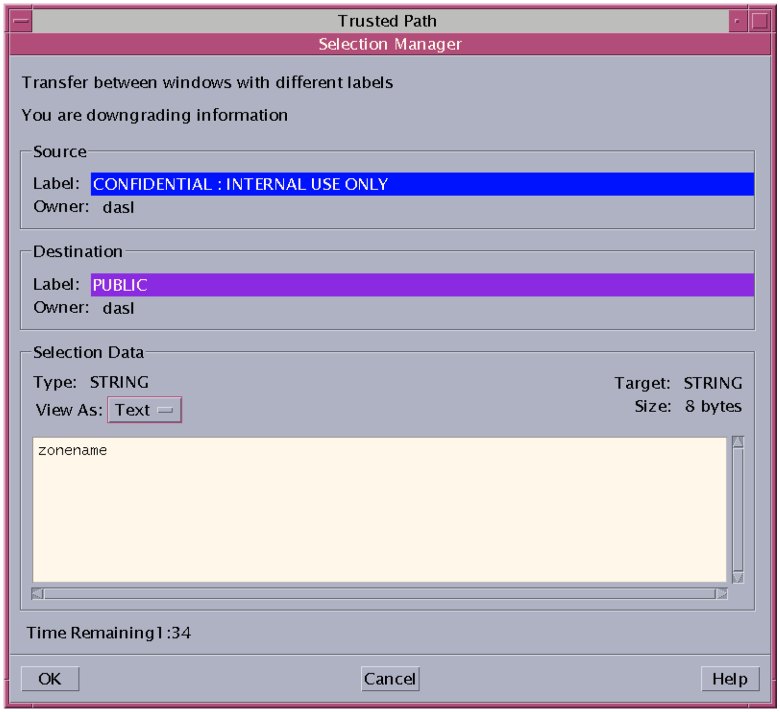
Different rules apply to selections within a window or file. Drag-and-drop of *selections* always requires equality of labels and ownership. Drag-and-drop between windows is mediated by the Selection Manager application, not by the `sel_config` file.

The rules for changing the label of selections are summarized in the following table.

TABLE 10–2 Conditions for Moving Selections to a New Label

Transaction Description	Label Relationship	Owner Relationship	Required Authorization
Copy and paste, or cut and paste of selections between windows	Same label	Same UID	None
	Downgrade	Same UID	`solaris.label.win.downgrade`
	Upgrade	Same UID	`solaris.label.win.upgrade`
	Downgrade	Different UIDs	`solaris.label.win.downgrade`
	Upgrade	Different UIDs	`solaris.label.win.upgrade`
Drag and drop of selections between windows	Same label	Same UID	None applicable

Trusted Extensions provides a selection confirmer to mediate label changes. This window appears when an authorized user attempts to change the label of a file or selection. The user has 120 seconds to confirm the operation. To change the security level of data without this window requires the `solaris.label.win.noview` authorization, in addition to the relabeling authorizations. The following illustration shows a selection, `zonename`, in the window.



By default, the selection confirmer displays whenever data is being transferred to a different label. If a selection requires several transfer decisions, the automatic reply mechanism provides a way to reply once to the several transfers. For more information, see the `sel_config(4)` man page and the following section.

`sel_config` File

The `sel_config` file is checked to determine the behavior of the selection confirmer when an operation would upgrade or downgrade a label.

The `sel_config` file defines the following:

- A list of selection types to which automatic replies are given
- Whether certain types of operations can be automatically confirmed
- Whether a selection confirmer dialog box is displayed

CHAPTER 11

Administering Security Requirements in Trusted Extensions (Tasks)

This chapter contains tasks that are commonly performed on a system that is configured with Solaris Trusted Extensions.

Common Tasks in Trusted Extensions (Task Map)

The following task map describes procedures that set up a working environment for administrators of Trusted Extensions.

Task	Description	For Instructions
Change the editor program for the trusted editor.	Specify the editor for administrative files.	"How to Assign the Editor of Your Choice as the Trusted Editor" on page 166
Change the password for root.	Specify a new password for the `root` user, or for the `root` role.	"How to Change the Password for `root`" on page 167
Change the password for a role.	Specifies a new password for your current role.	Example 11–2
Use the Secure Attention key combination.	Gets control of the mouse or keyboard. Also, tests whether the mouse or keyboard is trusted.	"How to Regain Control of the Desktop's Current Focus" on page 168
Determine the hexadecimal number for a label.	Displays the internal representation for a text label.	"How to Obtain the Hexadecimal Equivalent for a Label" on page 168
Determine the text representation for a label.	Displays the text representation for a hexadecimal label.	"How to Obtain a Readable Label From Its Hexadecimal Form" on page 170
Edit system files.	Securely edits Solaris or Trusted Extensions system files.	"How to Change Security Defaults in System Files" on page 170
Allocate a device.	Uses a peripheral device to add information to or remove information from the system.	"How to Allocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*

Task	Description	For Instructions
Administer a host remotely.	Administers Solaris or Trusted Extensions hosts from a remote host.	Chapter 14, "Remote Administration in Trusted Extensions (Tasks)"

▼ How to Assign the Editor of Your Choice as the Trusted Editor

The trusted editor uses the value of the `$EDITOR` environment variable as its editor.

Before You Begin You must be in a role in the global zone.

1. **Determine the value of the `$EDITOR` variable.**

```
# echo $EDITOR
```

The following are editor possibilities. The `$EDITOR` variable might also not be set.

- `/usr/bin/gedit` – Is the editor that GNOME provides. Trusted GNOME is the trusted version of that desktop.
- `/usr/bin/vi` – Is the visual editor.

2. **Set the value of the `$EDITOR` variable.**

- **To set the value permanently, modify the value in the shell initialization file for the role.**

  For example, in the role's home directory, modify the `.kshrc` file for a Korn shell, and the `.cshrc` file for a C shell.

- **To set the value for the current shell, set the value in the terminal window.**

  For example, in a Korn shell, use the following commands:

  ```
  # setenv EDITOR=pathname-of-editor
  # export $EDITOR
  ```

  In a C shell, use the following command:

  ```
  # setenv EDITOR=pathname-of-editor
  ```

  In a Bourne shell, use the following commands:

  ```
  # EDITOR=pathname-of-editor
  # export EDITOR
  ```

Example 11–1 Specifying the Editor for the Trusted Editor

The Security Administrator role wants to use `vi` when editing system files. A user who has assumed the role modifies the `.kshrc` initialization file in the role's home directory.

```
$ cd /home/secadmin
$ vi .kshrc

## Interactive shell
set -o vi
...
export EDITOR=vi
```

The next time that any user assumes the Security Administrator role, `vi` is the trusted editor.

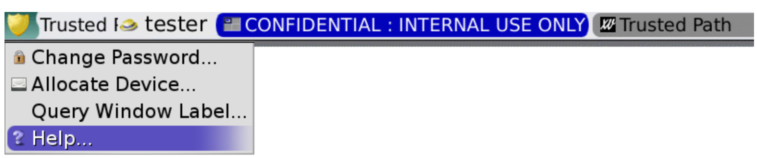
▼ How to Change the Password for `root`

The Security Administrator role is authorized to change any account's password at any time by using the Solaris Management Console. However, the Solaris Management Console cannot change the password of a system account. A *system account* is an account whose UID is below 100. `root` is a system account because its UID is 0.

1. **Become superuser.**

   If your site has made superuser into the `root` role, assume the `root` role.

2. **Choose Change Password from the trusted path menu.**



3. **Change the password, and confirm the change.**

Example 11–2 Changing the Password for a Role

Any user who can assume a role that is defined in LDAP can use the Trusted Path menu to change the password for the role. The password is then changed in LDAP for all users who attempt to assume the role.

As in the Solaris OS, the Primary Administrator role can change the password for a role by using the Solaris Management Console. In Trusted Extensions, the Security Administrator role can change another role's password by using the Solaris Management Console.

How to Regain Control of the Desktop's Current Focus

The “Secure Attention” key combination can be used to break a pointer grab or a keyboard grab by an untrusted application. The key combination can also be used to verify if a pointer or a keyboard has been grabbed by a trusted application. On a multiheaded system that has been spoofed to display more than one trusted stripe, this key combination warps the pointer to the authorized trusted stripe.

1. **To regain control of a Sun keyboard, use the following key combination.**

   Press the keys simultaneously to regain control of the current desktop focus. On the Sun keyboard, the diamond is the Meta key.

   ```
   <Meta> <Stop>
   ```

   If the grab, such as a pointer, is not trusted, the pointer moves to the stripe. A trusted pointer does not move to the trusted stripe.

2. **If you are not using a Sun keyboard, use the following key combination.**

   ```
   <Alt> <Break>
   ```

   Press the keys simultaneously to regain control of the current desktop focus on your laptop.

Example 11–3 Testing If the Password Prompt Can Be Trusted

On an x86 system that is using a Sun keyboard, the user has been prompted for a password. The cursor has been grabbed, and is in the password dialog box. To check that the prompt is trusted, the user presses the <Meta> <Stop> keys simultaneously. When the pointer remains in the dialog box, the user knows that the password prompt is trusted.

If the pointer had moved to the trusted stripe, the user would know that the password prompt could not be trusted, and contact the administrator.

Example 11–4 Forcing the Pointer to the Trusted Stripe

In this example, a user is not running any trusted processes but cannot see the mouse pointer. To bring the pointer to the center of the trusted stripe, the user presses the <Meta> <Stop> keys simultaneously.

How to Obtain the Hexadecimal Equivalent for a Label

This procedure provides an internal hexadecimal representation of a label. This representation is safe for storing in a public directory. For more information, see the `atohexlabel`(1M) man page.

Before You Begin You must be in the Security Administrator role in the global zone. For details, see “How to Enter the Global Zone in Trusted Extensions” on page 153.

- **To obtain a hexadecimal value for a label, do one of the following.**
  - **To obtain the hexadecimal value for a sensitivity label, pass the label to the command.**

    ```
    $ atohexlabel "CONFIDENTIAL : NEED TO KNOW"
    0x0004-08-68
    ```

  - **To obtain the hexadecimal value for a clearance, use the `-c` option.**

    ```
    $ atohexlabel -c "CONFIDENTIAL NEED TO KNOW"
    0x0004-08-68
    ```

Note – Human readable sensitivity labels and clearance labels are formed according to rules in the `label_encodings` file. Each type of label uses rules from a separate section of this file. When a sensitivity label and a clearance label both express the same underlying level of sensitivity, the labels have identical hexadecimal forms. However, the labels can have different human readable forms. System interfaces that accept human readable labels as input expect one type of label. If the text strings for the label types differ, these text strings cannot be used interchangeably.

In the default `label_encodings` file, the text equivalent of a clearance label does not include a colon (:).

Example 11–5 Using the `atohexlabel` Command

When you pass a valid label in hexadecimal format, the command returns the argument.

```
$ atohexlabel 0x0004-08-68
0x0004-08-68
```

When you pass an administrative label, the command returns the argument.

```
$ atohexlabel admin_high
ADMIN_HIGH
atohexlabel admin_low
ADMIN_LOW
```

Troubleshooting The error message `atohexlabel parsing error found in <string> at position 0` indicates that the `<string>` argument that you passed to `atohexlabel` was not a valid label or clearance. Check your typing, and check that the label exists in your installed `label_encodings` file.

How to Obtain a Readable Label From Its Hexadecimal Form

This procedure provides a way to repair labels that are stored in internal databases. For more information, see the hextoalabel(1M) man page.

Before You Begin You must be in the Security Administrator role in the global zone.

- **To obtain the text equivalent for an internal representation of a label, do one of the following.**
  - **To obtain the text equivalent for a sensitivity label, pass the hexadecimal form of the label.**

    ```
    $ hextoalabel 0x0004-08-68
    CONFIDENTIAL : NEED TO KNOW
    ```

  - **To obtain the text equivalent for a clearance, use the `-c` option.**

    ```
    $ hextoalabel -c 0x0004-08-68
    CONFIDENTIAL NEED TO KNOW
    ```

How to Change Security Defaults in System Files

In Trusted Extensions, the security administrator changes or accesses default security settings on a system.

Files in the `/etc/security` and `/etc/default` directories contain security settings. On a Solaris system, superuser can edit these files. For Solaris security information, see Chapter 3, "Controlling Access to Systems (Tasks)," in *System Administration Guide: Security Services*.

Caution – Relax system security defaults only if site security policy allows you to.

Before You Begin You must be in the Security Administrator role in the global zone.

- **Use the trusted editor to edit the system file.**

  For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

File	Task	For More Information
`/etc/default/login`	Reduce the allowed number of password tries.	See the example under "How to Monitor All Failed Login Attempts" in *System Administration Guide: Security Services*. passwd(1) man page

File	Task	For More Information
/etc/default/kbd	Disable keyboard shutdown.	"How to Disable a System's Abort Sequence" in *System Administration Guide: Security Services* **Note –** On hosts that are used by administrators for debugging, the default setting for KEYBOARD_ABORT allows access to the kadb kernel debugger. For more information about the debugger, see the kadb(1M) man page.
/etc/security/policy.conf	Require a more powerful algorithm for user passwords. Remove a basic privilege from all users of this host. Restrict users of this host to Basic Solaris User authorizations.	policy.conf(4) man page
/etc/default/passwd	Require users to change passwords frequently. Require users to create maximally different passwords. Require a longer user password. Require a password that cannot be found in your dictionary.	passwd(1) man page

◆ ◆ ◆ CHAPTER 12

Users, Rights, and Roles in Trusted Extensions (Overview)

This chapter describes essential decisions that you must make before creating regular users, and provides additional background information for managing user accounts. The chapter assumes that the initial setup team has set up roles and a limited number of user accounts. These users can assume the roles that are used to configure and administer Solaris Trusted Extensions. For details, see "Creating Roles and Users in Trusted Extensions" on page 79.

User Security Features in Trusted Extensions

Trusted Extensions software adds the following security features to users, roles, or rights profiles:

- A user has a label range within which the user can use the system.
- A role has a label range within which the role can be used to perform administrative tasks.
- Commands in a Trusted Extensions rights profile have a label attribute. The command must be performed within a label range, or at a particular label.
- Trusted Extensions software adds privileges and authorizations to the set of privileges and authorizations that are defined by the Solaris OS.

Administrator Responsibilities for Users

The System Administrator role creates user accounts. The Security Administrator role sets up the security aspects of an account.

If you are using the Sun Java™ System Directory Server for the LDAP naming service, check that the initial setup team configured the `tsol_ldap.tbx` toolbox. For the procedure, see "Configuring the Solaris Management Console for LDAP (Task Map)" on page 113.

For details on setting up users and roles, see the following:

- "How to Create the First Role (Primary Administrator)" in *System Administration Guide: Basic Administration*
- "Setting Up User Accounts (Task Map)" in *System Administration Guide: Basic Administration*
- Part III, "Roles, Rights Profiles, and Privileges," in *System Administration Guide: Security Services*

System Administrator Responsibilities for Users

In Trusted Extensions, the System Administrator role is responsible for determining who can access the system. The system administrator is responsible for the following tasks:

- Adding and deleting users
- Adding and deleting roles
- Modifying user and role configurations, other than security attributes

Security Administrator Responsibilities for Users

In Trusted Extensions, the Security Administrator role is responsible for all security attributes of a user or role. The security administrator is responsible for the following tasks:

- Assigning and modifying the security attributes of a user, role, or rights profile
- Creating and modifying rights profiles
- Assigning rights profiles to a user or role
- Assigning privileges to a user, role, or rights profile
- Assigning authorizations to a user, a role, or rights profile
- Removing privileges from a user, role, or rights profile
- Removing authorizations from a user, role, or rights profile

Typically, the Security Administrator role creates rights profiles. However, if a profile needs capabilities that the Security Administrator role cannot grant, then superuser or the Primary Administrator role can create the profile.

Before creating a rights profile, the security administrator needs to analyze whether any of the commands in the new profile need privilege or authorization to be successful. The man pages for individual commands list the privileges and authorizations that might be needed.

Decisions to Make Before Creating Users in Trusted Extensions

The following decisions affect what users are able to do in Trusted Extensions and how much effort is required. Some decisions are the same as the decisions that you would make when installing the Solaris OS. However, decisions that are specific to Trusted Extensions can affect site security and ease of use.

- Decide whether to change default user security attributes in the `policy.conf` file. User defaults in the `label_encodings` file were configured by the initial setup team. For a description of the defaults, see "Default User Security Attributes in Trusted Extensions" on page 175.
- Decide which startup files, if any, to copy or link from each user's minimum-label home directory to the user's higher-level home directories. For the procedure, see "How to Configure Startup Files for Users in Trusted Extensions" on page 184.
- Decide if users can access peripheral devices, such as the microphone, CD-ROM drive, and JAZ drive.

  If access is permitted to some users, decide if your site requires additional authorizations to satisfy site security. For the default list of device-related authorizations, see "How to Assign Device Authorizations" on page 341. For a finer-grained set of device authorizations, see "Customizing Device Authorizations in Trusted Extensions (Task Map)" on page 337.

Default User Security Attributes in Trusted Extensions

Settings in the `label_encodings` and the `policy.conf` files together define default security attributes for user accounts. The values that you explicitly set for a user override these system values. Some values that are set in these files also apply to role accounts. For security attributes that you can explicitly set, see "Configurable User Attributes in Trusted Extensions" on page 176.

`label_encodings` File Defaults

The `label_encodings` file defines a user's minimum label, clearance, and default label view. For details about the file, see the `label_encodings(4)` man page. Your site's `label_encodings` file was installed by your initial setup team. Their decisions were based on "Devising a Label Strategy" on page 29, and examples from *Solaris Trusted Extensions Label Administration*.

Label values that the security administrator explicitly sets for individual users in the Solaris Management Console are derived from the `label_encodings` file. Explicitly set values override the values in the `label_encodings` file.

`policy.conf` File Defaults in Trusted Extensions

The Solaris `/etc/security/policy.conf` file contains the default security settings for the system. Trusted Extensions adds two keywords to this file. You can add these keyword=value pairs to the file if you want to change the system-wide value. These keywords are enforced by Trusted Extensions.

TABLE 12–1 Trusted Extensions Security Defaults in `policy.conf` File

Keyword	Default Value	Possible Values	Notes
IDLECMD	LOCK	LOCK | LOGOUT	Does not apply to roles.
IDLETIME	30	0 to 120 minutes	Does not apply to roles.

The authorizations and rights profiles that are defined in the `policy.conf` file are *in addition* to any authorizations and profiles that are assigned to individual accounts. For the other fields, the individual user's value overrides the system value.

"Planning User Security in Trusted Extensions" on page 33 includes a table of every `policy.conf` keyword. See also the `policy.conf(4)` man page.

Configurable User Attributes in Trusted Extensions

The Solaris Management Console 2.1 is your tool for creating and modifying user accounts. For users who can log in at more than one label, you might also want to set up `.copy_files` and `.link_files` files in each user's minimum–label home directory.

The User Accounts tool in the Solaris Management Console works as it does in the Solaris OS, with two exceptions:

- Trusted Extensions adds attributes to user accounts.
- Home directory server access requires administrative attention in Trusted Extensions.
  - You create the home directory server entry the same as you do on a Solaris system.
  - Then, you and the user perform additional steps to mount the home directory at every user label.

As described in "How to Add a User With the Solaris Management Console's Users Tool" in *System Administration Guide: Basic Administration*, a wizard enables you to create user accounts quickly. After using the wizard, you can modify the user's default Trusted Extensions attributes.

For more information about the `.copy_files` and `.link_files` files, see "`.copy_files` and `.link_files` Files" on page 179.

Security Attributes That Must Be Assigned to Users

The Security Administrator role must specify some security attributes for new users, as the following table shows. For information about the files that contain default values, see "Default User Security Attributes in Trusted Extensions" on page 175.

TABLE 12–2 Security Attributes That Are Assigned After User Creation

User Attribute	Location of Default Value	Is Action Required	Effect of Action
Password	None	Required	User has password
Roles	None	Optional	User can assume a role
Authorizations	`policy.conf` file	Optional	User has additional authorizations
Rights Profiles	`policy.conf` file	Optional	User has additional rights profiles
Labels	`label_encodings` file	Optional	User has different default label or accreditation range
Privileges	`policy.conf` file	Optional	User has different set of privileges
Account Usage	`policy.conf` file	Optional	User has different setting for computer when it is idle
Audit	`audit_control` file	Optional	User is audited differently from the system audit settings

Security Attribute Assignment to Users in Trusted Extensions

The Security Administrator role assigns security attributes to users in the Solaris Management Console after the user accounts are created. If you have set up correct defaults, your next step is to assign security attributes only for users who need exceptions to the defaults.

When assigning security attributes to users, the security administrator considers the following information:

Assigning Passwords

The Security Administrator role assigns passwords to user accounts after the accounts have been created. After this initial assignment, users can change their passwords.

As in the Solaris OS, users can be forced to change their passwords at regular intervals. The password aging options limit how long any intruder who is able to guess or steal a password could potentially access the system. Also, establishing a minimum length of time to elapse before changing a password prevents a user with a new password from reverting immediately to the old password. For details, see the `passwd(1)` man page.

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

Assigning Roles

A user is not required to have a role. A single user can be assigned more than one role if doing so is consistent with your site's security policy.

Assigning Authorizations

As in the Solaris OS, assigning authorizations directly to a user adds those authorizations to existing authorizations. In Trusted Extensions, you add the authorizations to a rights profile, then assign the profile to the user.

Assigning Rights Profiles

As in the Solaris OS, the order of profiles is important. The profile mechanism uses the first instance of the command in an account's profile set.

You can use the sorting order of profiles to your advantage. If you want a command to run with different security attributes from those attributes that are defined for the command in an existing profile, create a new profile with the preferred assignments for the command. Then, insert that new profile before the existing profile.

Note – Do not assign rights profiles that include administrative commands to a regular user. The profile would not work because a regular user cannot enter the global zone.

Changing Privilege Default

The default privilege set can be too liberal for many sites. To restrict the privilege set for any regular user on a system, change the `policy.conf` file setting. To change the privilege set for individual users, use the Solaris Management Console. For an example, see "How to Restrict a User's Set of Privileges" on page 192.

Changing Label Defaults

Changing a user's label defaults creates an exception to the user defaults in the `label_encodings` file.

Changing Audit Defaults

As in the Solaris OS, assigning audit classes to a user creates exceptions to the audit classes that are assigned in the `/etc/security/audit_control` file on the system. For more information about auditing, see Chapter 24, "Trusted Extensions Auditing (Overview)."

`.copy_files` and `.link_files` Files

In Trusted Extensions, files are automatically copied from the skeleton directory *only* into the zone that contains the account's minimum label. To ensure that zones at higher labels can use startup files, either the user or the administrator must create the files `.copy_files` and `.link_files`.

The Trusted Extensions files `.copy_files` and `.link_files` help to automate the copying or linking of startup files into every label of an account's home directory. Whenever a user creates a workspace at a new label, the `updatehome` command reads the contents of `.copy_files` and `.link_files` at the account's minimum label. The command then copies or links every listed file into the higher-labeled workspace.

The `.copy_files` file is useful when a user wants a slightly different startup file at different labels. Copying is preferred, for example, when users use different mail aliases at different labels. The `.link-files` file is useful when a startup file should be identical at any label that it is invoked. Linking is preferred, for example, when one printer is used for all labeled print jobs. For example files, see "How to Configure Startup Files for Users in Trusted Extensions" on page 184.

The following lists some startup files that you might want users to be able to link to higher labels or to copy to higher labels:

`.acrorc`	`.login`	`.signature`
`.aliases`	`.mailrc`	`.soffice`
`.cshrc`	`.mime_types`	`.Xdefaults`
`.dtprofile`	`.newsrc`	`.Xdefaults-`*hostname*
`.emacs`	`.profile`	

CHAPTER 13

Managing Users, Rights, and Roles in Trusted Extensions (Tasks)

This chapter provides the Solaris Trusted Extensions procedures for configuring and managing users, user accounts, and rights profiles.

Customizing the User Environment for Security (Task Map)

The following task map describes common tasks that you can perform when customizing a system for all users, or when customizing an individual user's account.

Task	Description	For Instructions
Change label attributes.	Modify label attributes, such as minimum label and default label view, for a user account.	"How to Modify Default User Label Attributes" on page 182
Change Trusted Extensions policy for all users of a system.	Changes the `policy.conf` file.	"How to Modify `policy.conf` Defaults" on page 182
	Turns on the screensaver after a set amount of time. Logs the user out after a set amount of time that the system is idle.	Example 13–1
	Removes unnecessary privileges from all ordinary users of a system.	Example 13–2
	Prevents labels from being visible on a single-label system.	Example 13–3
	Removes labels from printed output at a public kiosk.	Example 13–4

Task	Description	For Instructions
Configure initialization files for users.	Configures startup files, such as `.cshrc`, `.copy_files`, and `.soffice` for all users.	"How to Configure Startup Files for Users in Trusted Extensions" on page 184
Lengthen the timeout for file relabeling.	Configures some applications to enable authorized users to relabel files.	"How to Lengthen the Timeout When Relabeling Information" on page 187
Log in to a failsafe session.	Fixes faulty user initialization files.	"How to Log In to a Failsafe Session in Trusted Extensions" on page 188

▼ How to Modify Default User Label Attributes

You can modify the default user label attributes during the configuration of the first system. The changes must be copied to every Trusted Extensions host.

Before You Begin You must be in the Security Administrator role in the global zone. For details, see "How to Enter the Global Zone in Trusted Extensions" on page 153.

1. **Review the default user attribute settings in the `/etc/security/tsol/label_encodings` file.**

   For the defaults, see "`label_encodings` File Defaults" on page 175.

2. **Modify the user attribute settings in the `label_encodings` file.**

   Use the trusted editor. For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

   The `label_encodings` file should be the same on all hosts.

3. **Distribute a copy of the file to every Trusted Extensions host.**

▼ How to Modify `policy.conf` Defaults

Changing the `policy.conf` defaults in Trusted Extensions is similar to changing any security-relevant system file in the Solaris OS. In Trusted Extensions, you use a trusted editor to modify system files.

Before You Begin You must be in the Security Administrator role in the global zone. For details, see "How to Enter the Global Zone in Trusted Extensions" on page 153.

1. **Review the default settings in the `/etc/security/policy.conf` file.**

   For Trusted Extensions keywords, see Table 12–1.

2 Modify the settings.

Use the trusted editor to edit the system file. For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

Example 13–1 Changing the System's Idle Settings

In this example, the security administrator wants idle systems to return to the login screen. The default locks an idle system. Therefore, the Security Administrator role adds the `IDLECMD` keyword=value pair to the `/etc/security/policy.conf` file as follows:

```
IDLECMD=LOGOUT
```

The administrator also wants systems to be idle a shorter amount of time before logout. Therefore, the Security Administrator role adds the `IDLETIME` keyword=value pair to the `policy.conf` file as follows:

```
IDLETIME=10
```

The system now logs out the user after the system is idle for 10 minutes.

Example 13–2 Modifying Every User's Basic Privilege Set

In this example, the security administrator of a Sun Ray™ installation does not want regular users to view the processes of other Sun Ray users. Therefore, on every system that is configured with Trusted Extensions, the administrator removes `proc_info` from the basic set of privileges. The `PRIV_DEFAULT` setting in the `/etc/policy.conf` file is modified as follows:

```
PRIV_DEFAULT=basic,!proc_info
```

Example 13–3 Hiding Labels on a System

In this example, the security administrator changes the default setting in a system's `policy.conf` file to hide labels. Any user on this system would not view labels, unless the user was specifically configured to be able to view labels. This setting is reasonable on a single-label system, or on a system that is available to the general public.

```
# /etc/security/policy.conf
...
LABELVIEW=hidesl
```

To configure a user to override this setting, see "How to Hide Labels From a User" on page 194.

Example 13–4 Assigning Printing-Related Authorizations to All Users of a System

In this example, the security administrator enables a public kiosk computer to print without labels by typing the following in the computer's `/etc/security/policy.conf` file. At the next boot, print jobs by all users of this kiosk print without page labels.

```
AUTHS_GRANTED= solaris.print.unlabeled
```

Then, the administrator decides to save paper by removing banner and trailer pages. She first ensures that the Always Print Banners checkbox in the Print Manager is not selected. She then modifies the `policy.conf` entry to read the following and reboots. Now, all print jobs are unlabeled, and have no banner or trailer pages.

```
AUTHS_GRANTED= solaris.print.unlabeled,solaris.print.nobanner
```

▼ How to Configure Startup Files for Users in Trusted Extensions

Users can put a `.copy_files` file and `.link_files` file into their home directory at the label that corresponds to their minimum sensitivity label. Users can also modify the existing `.copy_files` and `.link_files` files at the users' minimum label. This procedure is for the administrator role to automate the setup for a site.

Before You Begin You must be in the System Administrator role in the global zone. For details, see "How to Enter the Global Zone in Trusted Extensions" on page 153.

1. **Create two Trusted Extensions startup files.**

   You are going to add `.copy_files` and `.link_files` to your list of startup files.

   ```
   # cd /etc/skel
   # touch .copy_files .link_files
   ```

2. **Customize the `.copy_files` file.**

   a. **Start the trusted editor.**

      For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

   b. **Type the full pathname to the `.copy_files` file.**

      ```
      /etc/skel/.copy_files
      ```

   c. **Type into `.copy_files`, one file per line, the files to be copied into the user's home directory at all labels.**

      Use "`.copy_files` and `.link_files` Files" on page 179 for ideas. For sample files, see Example 13–5.

3 Customize the `.link_files` file.

a. Type the full pathname to the `.link_files` file in the trusted editor.

`/etc/skel/.link_files`

b. Type into `.link_files`, one file per line, the files to be linked into the user's home directory at all labels.

4 Customize the other startup files for your users.

- For a discussion of what to include in startup files, see "Customizing a User's Work Environment" in *System Administration Guide: Basic Administration*.
- For details, see "How to Customize User Initialization Files" in *System Administration Guide: Basic Administration*.
- For an example, see Example 13–5.

5 (Optional) Create a `skelP` subdirectory for users whose default shell is a profile shell.

The `P` indicates the Profile shell.

6 Copy the customized startup files into the appropriate skeleton directory.

7 Use the appropriate `skel`*X* pathname when you create the user.

The *X* indicates the letter that begins the shell's name, such as `B` for Bourne, `K` for Korn, `C` for a C shell, and `P` for Profile shell.

Example 13–5 Customizing Startup Files for Users

In this example, the security administrator configures files for every user's home directory. The files are in place before any user logs in. The files are at the user's minimum label. At this site, the users' default shell is the C shell.

The security administrator creates a `.copy_files` and a `.link_files` file in the trusted editor with the following contents:

```
## .copy_files for regular users
## Copy these files to my home directory in every zone
.mailrc
.mozilla
.soffice
:wq

## .link_files for regular users with C shells
## Link these files to my home directory in every zone
.cshrc
```

```
.login
.Xdefaults
.Xdefaults-hostname
:wq

## .link_files for regular users with Korn shells
# Link these files to my home directory in every zone
.ksh
.profile
.Xdefaults
.Xdefaults-hostname
:wq
```

In the shell initialization files, the administrator ensures that the users' print jobs go to a labeled printer.

```
## .cshrc file
setenv PRINTER conf-printer1
setenv LPDEST  conf-printer1

## .ksh file
export PRINTER conf-printer1
export LPDEST  conf-printer1
```

The administrator modifies the `.Xdefaults-`*home-directory-server* file to force the `dtterm` command to source the `.profile` file for a new terminal.

```
## Xdefaults-HDserver
Dtterm*LoginShell: true
```

The customized files are copied to the appropriate skeleton directory.

```
$ cp .copy_files .link_files .cshrc .login .profile \
.mailrc .Xdefaults .Xdefaults-home-directory-server \
/etc/skelC
$ cp .copy_files .link_files .ksh .profile \
.mailrc .Xdefaults .Xdefaults-home-directory-server \
/etc/skelK
```

Troubleshooting

If you create a `.copy_files` files at your lowest label, then log in to a higher zone to run the `updatehome` command and the command fails with an access error, try the following:

- Verify that from the higher-level zone you can view the lower-level directory.

  ```
  higher-level zone# ls /zone/lower-level-zone/home/username
  ACCESS ERROR: there are no files under that directory
  ```

- If you cannot view the directory, then restart the automount service in the higher-level zone:

  ```
  higher-level zone# svcadm restart autofs
  ```

Unless you are using NFS mounts for home directories, the automounter in the higher-level zone should be loopback mounting from `/zone/`*lower-level-zone*`/export/home/`*username* to `/zone/`*lower-level-zone*`/home/`*username*.

▼ How to Lengthen the Timeout When Relabeling Information

In Trusted Extensions, the Selection Manager mediates transfers of information between labels. The Selection Manager appears for drag-and-drop operations, and for cut-and-paste operations. Some applications require that you set a suitable timeout so that the Selection Manager has time to intervene. A value of two minutes is sufficient.

Caution – Do not change the default timeout value on an unlabeled system. The operations fail with the longer timeout value.

Before You Begin You must be in the System Administrator role in the global zone. For details, see "How to Enter the Global Zone in Trusted Extensions" on page 153.

1 For the StarOffice™ application, do the following:

a. Navigate to the file *office-install-directory*`/VCL.xcu`**.**

where *office-install-directory* is the StarOffice installation directory, for example:

office-top-dir`/share/registry/data/org/staroffice`

b. Change the `SelectionTimeout` **property value to 120.**

Use the trusted editor. For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

The default value is three seconds. A value of 120 sets the timeout to two minutes.

2 For users of applications that rely on the GNOME ToolKit (GTK) library, change the selection timeout property value to two minutes.

Note – As an alternative, you could have each user change the selection timeout property value.

Most Sun Java™ Desktop System applications use the GTK library. Web browsers such as Mozilla, Firefox, and Thunderbird use the GTK library.

By default, the selection timeout value is 300, or five seconds. A value of 7200 sets the timeout to two minutes.

a. **Create a GTK startup file.**

   Name the file `.gtkrc-mine`. The `.gtkrc-mine` file belongs in the user's home directory at the minimum label.

b. **Add the selection timeout value to the file.**

```
## $HOME/.gtkrc-mine file
*gtk-selection-timeout: 7200
```

   As in the Solaris OS, the `gnome-settings-daemon` reads this file on startup.

3 **(Optional) Add the `.gtkrc-mine` file to the list in each user's `.link_files` file.**

For details, see "How to Configure Startup Files for Users in Trusted Extensions" on page 184.

▼ How to Log In to a Failsafe Session in Trusted Extensions

In Trusted Extensions, failsafe login is protected. If a regular user has customized shell initialization files and now cannot log in, you can use failsafe login to fix the user's files.

Before You Begin You must know the `root` password.

1 **As in the Solaris OS, choose Options –> Failsafe Session on the login screen.**

2 **At the prompt, have the user provide the user name and password.**

3 **At the prompt for the `root` password, provide the password for `root`.**

You can now debug the user's initialization files.

Managing Users and Rights With the Solaris Management Console (Task Map)

In Trusted Extensions, you must use the Solaris Management Console to administer users, authorizations, rights, and roles. To manage users and their security attributes, assume the Security Administrator role.

Task	Description	For Instructions
Modify a user's label range.	Modifies the labels at which a user can work. Modifications can restrict or extend the range that the `label_encodings` file permits.	"How to Modify a User's Label Range in the Solaris Management Console" on page 189
Create a rights profile for convenient authorizations.	Several authorizations exist that might be useful for regular users. Creates a profile for users who qualify to have these authorizations.	"How to Create a Rights Profile for Convenient Authorizations" on page 190
Modify a user's default privilege set.	Removes a privilege from the user's default privilege set.	"How to Restrict a User's Set of Privileges" on page 192
Prevent account locking for particular users.	Users who can assume a role must have account locking turned off.	"How to Prevent Account Locking for Users" on page 194
Hide labels on a user's screen.	On a single-label system, you might want a user to not view labels.	"How to Hide Labels From a User" on page 194
Enable a user to relabel data.	Authorizes a user to downgrade information or upgrade information.	"How to Enable a User to Change the Security Level of Data" on page 195
Remove a user from the system.	Completely removes a user and the user's processes..	"How to Delete a User Account From a Trusted Extensions System" on page 196
Handle other tasks.	Uses the Solaris Management Console to handle tasks that are not specific to Trusted Extensions.	"Handling Other Tasks in the Solaris Management Console (Task Map)" on page 197

▼ How to Modify a User's Label Range in the Solaris Management Console

You might want to extend a user's label range to give the user read access to an administrative application. For example, a user who can log in to the global zone could then run the Solaris Management Console. The user could view, but not not change the contents.

Alternatively, you might want to restrict the user's label range. For example, a guest user might be limited to one label.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Open a Trusted Extensions toolbox in the Solaris Management Console.**

   Use a toolbox of the appropriate scope. For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

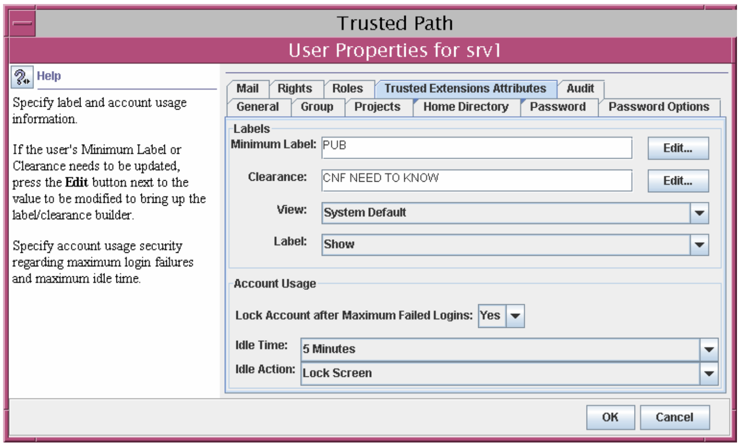
2. **Under System Configuration, navigate to User Accounts.**

   A password prompt might be displayed.

3 **Type the role password.**

4 **Select the individual user from User Accounts.**

5 **Click the Trusted Extensions Attributes tab.**



- **To extend the user's label range, choose a higher clearance.**

  You can also lower the minimum label.

- **To restrict the label range to one label, make the clearance equal to the minimum label.**

6 **To save the changes, click OK.**

▼ How to Create a Rights Profile for Convenient Authorizations

Where site security policy permits, you might want to create a rights profile that contains authorizations for users who can perform tasks that require authorization. To enable every user of a particular system to be authorized, see "How to Modify `policy.conf` Defaults" on page 182.

Before You Begin You must be in the Security Administrator role in the global zone.

1 **Open a Trusted Extensions toolbox in the Solaris Management Console.**

Use a toolbox of the appropriate scope. For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

2 Under System Configuration, navigate to Rights.

A password prompt might be displayed.

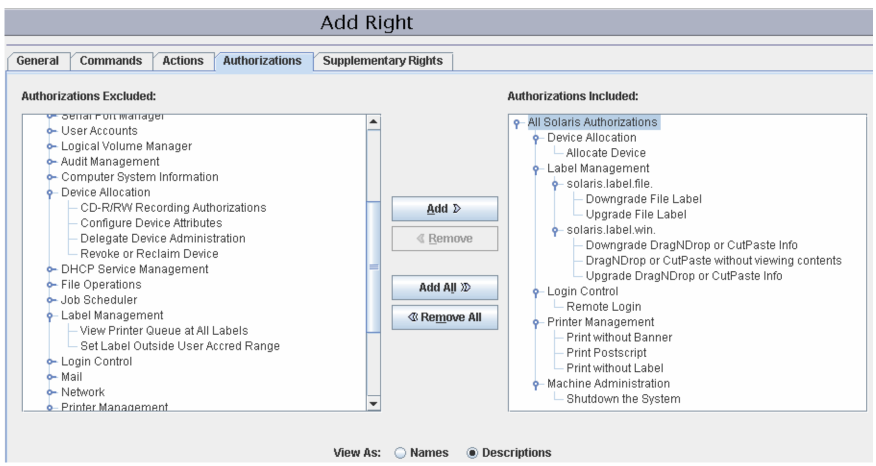
3 Type the role password.

4 To add a rights profile, click Action –> Add Right.

5 Create a rights profile that contains one or more of the following authorizations.

For the step-by-step procedure, see "How to Create or Change a Rights Profile" in *System Administration Guide: Security Services*.

In the following figure, the Authorizations Included window shows the authorizations that might be convenient for users.



- Allocate Device – Authorizes a user to allocate a peripheral device, such as a microphone.

  By default, Solaris users can read and write to a CD-ROM. However, in Trusted Extensions, only users who can allocate a device can access the CD-ROM drive. To allocate the drive for use requires authorization. Therefore, to read and write to a CD-ROM in Trusted Extensions, a user needs the Allocate Device authorization.
- Downgrade DragNDrop or CutPaste Info – Authorizes a user to select information from a higher-level file and place that information in a lower-level file.
- Downgrade File Label – Authorizes a user to lower the security level of a file
- DragNDrop or CutPaste without viewing contents – Authorizes a user to move information without viewing the information that is being moved.
- Print Postscript – Authorizes a user to print PostScript™ files.
- Print without Banner - Authorizes a user to print hard copy without a banner page.

- Print without Label – Authorizes a user to print hard copy that does not display labels.
- Remote Login – Authorizes a user to remotely log in.
- Shutdown the System – Authorizes a user to shut down the system and to shut down a zone.
- Upgrade DragNDrop or CutPaste Info – Authorizes a user to select information from a lower-level file and place that information in a higher-level file.
- Upgrade File Label – Authorizes a user to heighten the security level of a file.

6 **Assign the rights profile to a user or a role.**

For assistance, see the online help. For the step-by-step procedure, see "How to Change the RBAC Properties of a User" in *System Administration Guide: Security Services*.

Example 13–6 Assigning a Printing-Related Authorization to a Role

In the following example, the Security Administrator allows a role to print jobs without labels on body pages.

In the Solaris Management Console, the security administrator navigates to Administrative Roles. She views the rights profiles that are included in a particular role, then ensures that the print-related authorizations are contained in one of the role's rights profiles.

▼ How to Restrict a User's Set of Privileges

Site security might require that users be permitted fewer privileges than users are assigned by default. For example, at a site that uses Trusted Extensions on Sun Ray systems, you might want to prevent users from viewing other users' processes on the Sun Ray server.

Before You Begin You must be in the Security Administrator role in the global zone.

1 **Open a Trusted Extensions toolbox in the Solaris Management Console.**

Use a toolbox of the appropriate scope. For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

2 **Under System Configuration, navigate to User Accounts.**

A password prompt might be displayed.

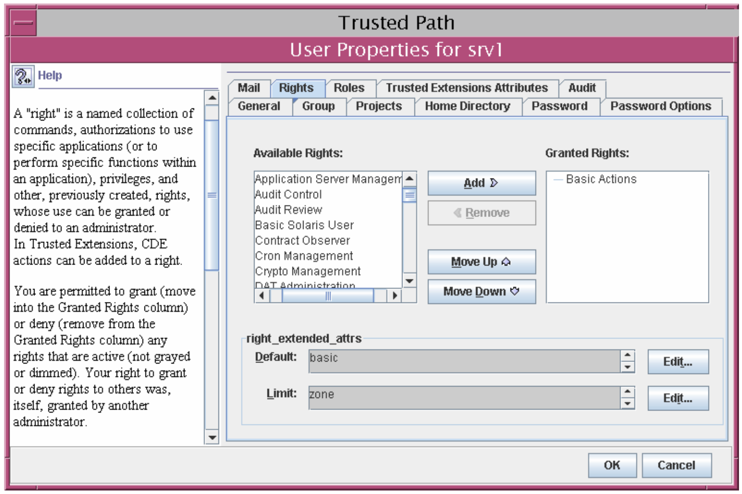
3 **Type the role password.**

4 **Double–click the icon for the user.**

5 **Remove one or more of the privileges in the `basic` set.**

a. **Double-click the icon for the user.**

b. Click the Rights tab.

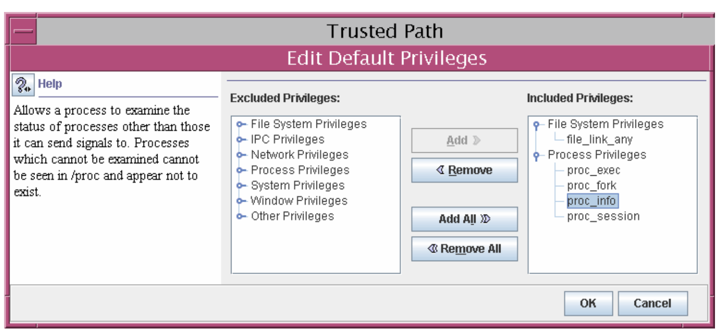


c. Click the Edit button to the right of the `basic` **set in the** `right_extended_attr` **field.**

d. Remove `proc_session` **or** `file_link_any`**.**

By removing the `proc_session` privilege, you prevent the user from examining any processes outside the user's current session. By removing the `file_link_any` privilege, you prevent the user from making hard links to files that are not owned by the user.

Caution – Do not remove the `proc_fork` or the `proc_exec` privilege. Without these privileges, the user would not be able to use the system.



6 To save the changes, click OK.

How to Prevent Account Locking for Users

Trusted Extensions extends the user security features in the Solaris Management Console to include account locking. Turn off account locking for users who can assume a role.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Start the Solaris Management Console.**

   Use a toolbox of the appropriate scope. For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

2. **Under System Configuration, navigate to User Accounts.**

   A password prompt might be displayed.

3. **Type the role password.**

4. **Double–click the icon for the user.**

5. **Click the Trusted Extensions Attributes tab.**

6. **In the Account Usage section, choose `No` from the pull-down menu next to `Lock account after maximum failed logins`.**

7. **To save the changes, click OK.**

How to Hide Labels From a User

Hiding labels is useful at a site where users can work at a single label only. An organization might not want regular users to see labels or to be aware of mandatory access controls. Ordinary users can then work whose desktop closely resembles the desktop on a Solaris system.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Open a Trusted Extensions toolbox in the Solaris Management Console.**

   Use a toolbox of the appropriate scope. For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

2. **Under System Configuration, navigate to User Accounts.**

   A password prompt might be displayed.

3. **Type the role password.**

4 **Double–click the icon for the user.**

5 **Click the Trusted Extensions Attributes tab.**

6 **Choose Hide from the Label: selection list.**

This setting overrides the value of `LABELVIEW` in the system's `policy.conf` file. For details, see "Default User Security Attributes in Trusted Extensions" on page 175.

7 **To save the changes, click OK.**

▼ How to Enable a User to Change the Security Level of Data

A regular user or a role can be authorized to change the security level, or labels, of files and directories. The user or role, in addition to having the authorization, must be configured to work at more than one label. And, the labeled zones must be configured to permit relabeling. For the procedure, see "How to Enable Files to be Relabeled From a Labeled Zone" on page 226.

Caution – Changing the security level of data is a privileged operation. This task is for trustworthy users only.

Before You Begin You must be in the Security Administrator role in the global zone.

1 **Follow the procedure "How to Create a Rights Profile for Convenient Authorizations" on page 190 to create a rights profile.**

The following authorizations enable a user to relabel a file:

- Downgrade File Label
- Upgrade File Label

The following authorizations enable a user to relabel information within a file:

- Downgrade DragNDrop or CutPaste Info
- DragNDrop or CutPaste Info Without Viewing
- Upgrade DragNDrop or CutPaste Info

2 **Use the Solaris Management Console to assign the profile to the appropriate users and roles.**

For assistance, use the online help. For a step-by-step procedure, see "How to Change the RBAC Properties of a User" in *System Administration Guide: Security Services*.

▼ How to Delete a User Account From a Trusted Extensions System

When a user is removed from the system, you must ensure that the user's home directory and any objects that the user owns are also deleted. As an alternative to deleting objects that are owned by the user, you might change the ownership of these objects to a valid user.

You must also ensure that all batch jobs that are associated with the user are also deleted. No objects or processes belonging to a removed user can remain on the system.

Before You Begin You must be in the System Administrator role.

1. **Archive the user's home directory at every label.**
2. **Archive the user's mail files at every label.**
3. **In the Solaris Management Console, delete the user account.**
   a. **Open a Trusted Extensions toolbox in the Solaris Management Console.**

      Use a toolbox of the appropriate scope. For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

   b. **Under System Configuration, navigate to User Accounts.**

      A password prompt might be displayed.

   c. **Type the role password.**

   d. **Select the user account to be removed, and click the Delete button.**

      You are prompted to delete the user's home directory and mail files. When you accept the prompt, the user's home directory and mail files are deleted in the global zone only.
4. **In every labeled zone, manually delete the user's directories and mail files.**

Note – You are responsible for finding and deleting the user's temporary files at all labels, such as files in `/tmp` directories.

Handling Other Tasks in the Solaris Management Console (Task Map)

Follow Solaris procedures to handle tasks in the Solaris Management Console. You must be superuser, or in a role in the global zone.

Task	For Instructions
Perform administrative tasks by using the Solaris Management Console.	Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*
Create users.	"Using the Solaris Management Tools With RBAC (Task Map)" in *System Administration Guide: Basic Administration*
Create roles.	"How to Create and Assign a Role by Using the GUI" in *System Administration Guide: Security Services*
Modify roles.	"How to Change the Properties of a Role" in *System Administration Guide: Security Services*
Create or modify a rights profile.	"How to Create or Change a Rights Profile" in *System Administration Guide: Security Services*
Change other security attributes of a user.	"How to Change the RBAC Properties of a User" in *System Administration Guide: Security Services*
Audit the actions of a role.	"How to Audit Roles" in *System Administration Guide: Security Services*
List the rights profiles by using `smprofile list -D`*name-service-type*`:/`*server-name*`/`*domain-name*	*Chapter 9, "Using Role-Based Access Control (Tasks)," in System Administration Guide: Security Services* or the `smprofile`(1M) man page

CHAPTER 14

Remote Administration in Trusted Extensions (Tasks)

This chapter describes how to use Trusted Extensions administrative tools to administer a remote system.

Secure Remote Administration in Trusted Extensions

By default, Trusted Extensions does not allow remote administration. Remote administration would present a significant security risk if users on remote untrusted systems could administer systems that are configured with Trusted Extensions. Therefore, systems are initially installed without the option of being remotely administered.

Until the network is configured, all remote hosts are assigned the `admin_low` security template. Therefore, the CIPSO protocol is not used or accepted for any connections. While in this initial state, systems are protected from remote attacks by several mechanisms. Mechanisms include `netservices` settings, default login policy, and PAM policy.

- When the `netservices` Service Management Facility (SMF) profile is set to `limited`, no remote services except secure shell are enabled. However, the `ssh` service cannot be used for remote logins because of the login and PAM policies.
- The `root` account cannot be used for remote logins because the default policy for `CONSOLE` in the `/etc/default/login` file prevents remote logins by `root`.
- Two PAM settings also affect remote logins.

  The `pam_roles` module always rejects local logins from accounts of type `role`. By default, this module also rejects remote logins. However, the system can be configured to accept remote logins by specifying `allow_remote` in the system's `pam.conf` entry.

Additionally, the `pam_tsol_account` module rejects remote logins into the global zone unless the CIPSO protocol is used. The intent of this policy is for remote administration to be performed by another Trusted Extensions system.

To enable remote login functionality, both systems must assign their peer to a CIPSO security template. If this approach is not practical, the network protocol policy can be relaxed by specifying the `allow_unlabeled` option in the `pam.conf` file. If either policy is relaxed, the default network template must be changed so that arbitrary machines cannot access the global zone. The `admin_low` template should be used sparingly, and the `tnrhdb` database should be modified so that the wildcard address `0.0.0.0` does not default to the `ADMIN_LOW` label. For details, see "Administering Trusted Extensions Remotely (Task Map)" on page 202 and "How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 279.

Methods for Administering Remote Systems in Trusted Extensions

Typically, administrators use the `rlogin` and `ssh` commands to administer remote systems from the command line. The Solaris Management Console can also be used. Also, a virtual networking computer (vnc) can be used to remotely display a multilevel desktop.

The following methods of remote administration are possible in Trusted Extensions:

- The root user can log in to a remote host from a terminal. See "How to Log In Remotely From the Command Line in Trusted Extensions" on page 202. This method works as it does on a Solaris system. This method is insecure.
- A role can log in to a remote host from a terminal in the role workspace. See "How to Log In Remotely From the Command Line in Trusted Extensions" on page 202.
- Administrators can start a Solaris Management Console server that is running on a remote system. See "How to Remotely Administer Systems by Using the Solaris Management Console From a Trusted Extensions System" on page 203.
- A user can log in to a remote multilevel desktop by using a vnc client program to connect to the Xvnc server on a Trusted Extensions system. See "How to Use Xvnc to Remotely Access a Trusted Extensions System" on page 207.

Remote Login by a Role in Trusted Extensions

As in the Solaris OS, a setting in the `/etc/default/login` file on each host must be changed to allow remote logins. Additionally, the `pam.conf` file might need to be modified. In Trusted Extensions, the security administrator is responsible for the change. For the procedures, see "Enable Remote Login by `root` User in Trusted Extensions" on page 120 and "Enable Remote Login by a Role in Trusted Extensions" on page 121.

On both Trusted Extensions and Solaris hosts, remote logins might or might not require authorization. "Remote Login Management in Trusted Extensions" on page 201 describes the conditions and types of logins that require authorization. By default, roles have the Remote Login authorization.

Remote Role-Based Administration From Unlabeled Hosts

In Trusted Extensions, users assume roles through the Trusted Path menu. The roles then operate in trusted workspaces. By default, roles cannot be assumed outside of the trusted path. If site policy permits, the security administrator can change the default policy. Administrators of unlabeled hosts that are running Solaris Management Console 2.1 client software can then administer trusted hosts.

- To change the default policy, see "Enable Remote Login by a Role in Trusted Extensions" on page 121.
- To administer systems remotely, see "How to Log In Remotely From the Command Line in Trusted Extensions" on page 202.

This policy change only applies when the user on the remote unlabeled system has a user account on the Trusted Extensions host. The Trusted Extensions user must have the ability to assume an administrative role. The role can then use the Solaris Management Console to administer the remote system.

Caution – If remote administration from a non-Trusted Extensions host is enabled, the administrative environment is less protected than a Trusted Extensions administrative workspace. Be cautious when typing passwords and other secure data. As a precaution, shut down all untrusted applications before starting the Solaris Management Console.

Remote Login Management in Trusted Extensions

A remote login between two Trusted Extensions hosts is considered to be an extension of the current login session.

An authorization is not required when the `rlogin` command does not prompt for a password. If an `/etc/hosts.equiv` file or a `.rhosts` file in the user's home directory on the remote host lists either the username or the host from which the remote login is being attempted, no password is required. For more information, see the `rhosts(4)` and `rlogin(1)` man pages.

For all other remote logins, including logins with the `ftp` command, the Remote Login authorization is required.

To create a rights profile that includes the Remote Login authorization, see "Managing Users and Rights With the Solaris Management Console (Task Map)" on page 188.

Administering Trusted Extensions Remotely (Task Map)

The following task map describes the tasks used to administer a remote Trusted Extensions system.

Task	Description	For Instructions
Enable `root` to remotely log in to a Trusted Extensions system.	Enables the `root` user to work remotely from a labeled system.	"Enable Remote Login by `root` User in Trusted Extensions" on page 120
Enable a role to remotely log in to a Trusted Extensions system.	Allows any role to work remotely from a labeled system.	"Enable Remote Login by a Role in Trusted Extensions" on page 121
Enable remote login from an unlabeled system to a Trusted Extensions system.	Allows any user or role to work remotely from an unlabeled system.	"Enable Remote Login From an Unlabeled System" on page 123
Log in remotely to a Trusted Extensions system.	Logs in as a role to a Trusted Extensions system.	"How to Log In Remotely From the Command Line in Trusted Extensions" on page 202
Administer a system remotely.		
	From a Trusted Extensions system, uses the Solaris Management Console to administer the remote host.	"How to Remotely Administer Systems by Using the Solaris Management Console From a Trusted Extensions System" on page 203
	From an unlabeled system, uses the Solaris Management Console to administer remote Trusted Extensions hosts.	"How to Remotely Administer Systems by Using the Solaris Management Console From an Unlabeled System" on page 204
Administer and use a remote system	From any client, uses the Xvnc server on the remote Trusted Extensions to display a multilevel session back to the client	"How to Use Xvnc to Remotely Access a Trusted Extensions System" on page 207
Enable specific users to log in to the global zone.	Uses user and network tools in the Solaris Management Console to enable specific users to access the global zone.	"How to Enable Specific Users to Log In Remotely to the Global Zone in Trusted Extensions" on page 206

▼ How to Log In Remotely From the Command Line in Trusted Extensions

---

Note – The `telnet` command cannot be used for remote role assumption because this command cannot pass the primary and role identities to the `pam_roles` module.

Before You Begin The user and the role must be identically defined on the local and the remote system.

The role must have the Remote Login authorization. By default, this authorization is in the Remote Administration, and the Maintenance and Repair rights profiles.

The security administrator has completed the procedure "Enable Remote Login by a Role in Trusted Extensions" on page 121 on every system that can be remotely administered. If the system can be administered from an unlabeled system, the procedure "Enable Remote Login From an Unlabeled System" on page 123 has also been completed.

- **From the workspace of a user who can assume a role, log in to the remote host.**

  Use the `rlogin` command, the `ssh` command, or the `ftp` command.

  - If the `rlogin -l` or `ssh` command is used to log in, all commands that are in the role's rights profiles are available.
  - If the `ftp` command is used, see the `ftp(1)` man page for the commands that are available.

How to Remotely Administer Systems by Using the Solaris Management Console From a Trusted Extensions System

The Solaris Management Console provides a remote administration interface to manage users, rights, roles, and the network. You assume a role to use the Console. In this procedure, you run the Console on the local system and specify the remote system as the server.

Before You Begin You have completed the following procedures:

- On both systems – "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60
- On the remote system – "Enable Remote Login by a Role in Trusted Extensions" on page 121 and "Enable the Solaris Management Console to Accept Network Communications" on page 114
- On the remote system that is the LDAP server – "Configuring the Solaris Management Console for LDAP (Task Map)" on page 113

1. **On the local system, log in as the user who is defined identically on the remote system.**

2. **Assume the role that you plan to use to administer the system.**

3 **In the role, start the Solaris Management Console.**

For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

a. **In the Server dialog box, type the name of the remote server.**

- **If you are using LDAP as a naming service, type the name of the LDAP server.**

  Then, choose one of the following scopes.

  - **To administer the databases in the naming service, choose the `Scope=LDAP` toolbox.**

    `This Computer (`*ldap-server*`: Scope=LDAP, Policy=TSOL)`

  - **To administer the local files on the LDAP server, choose the `Scope=Files` toolbox.**

    `This Computer (`*ldap-server*`: Scope=Files, Policy=TSOL)`

- **If you are not using LDAP as a naming service, type the name of the remote system that you want to administer.**

  Then, choose the `Scope=Files` toolbox.

  `This Computer (`*remote-system*`: Scope=Files, Policy=TSOL)`

4 **Select a tool under System Configuration.**

When you select a tool such as User, a dialog box displays the Solaris Management Console server name, your user name, your role name, and a place to type the role's password. Make sure that the entries are correct.

5 **In the role that is defined identically on the local and the remote systems, log in to the Solaris Management Console server.**

Type the role's password and press Login as Role. You can now use the Solaris Management Console to manage the system.

▼ How to Remotely Administer Systems by Using the Solaris Management Console From an Unlabeled System

In this procedure, you run the Solaris Management Console client and server on the remote system, and display the Console on the local system.

Before You Begin The Trusted Extensions system must have assigned the label `ADMIN_LOW` to the local system.

Note – A system that is not running the CIPSO protocol, such as a Trusted Solaris system, is an unlabeled system from the viewpoint of a Trusted Extensions system.

The Solaris Management Console server on the remote system must be configured to accept the remote connection. For the procedure, see "Enable the Solaris Management Console to Accept Network Communications" on page 114.

Both systems must have the same user who is assigned the same role that can use the Solaris Management Console. The user can have the normal user's label range, but the role must have the range from `ADMIN_LOW` to `ADMIN_HIGH`.

You must be in an administrative role in the global zone.

1. **Enable the local X server to display the remote Solaris Management Console.**

   ```
   # xhost + TX-SMC-Server
   # echo $DISPLAY
   :n.n
   ```

2. **On the local system, become the user who can assume a role for the Solaris Management Console.**

   ```
   # su - same-username-on-both-systems
   ```

3. **As that user, log in to the remote server as the role.**

   ```
   $ rlogin -l same-rolename-on-both-systems  TX-SMC-Server
   ```

4. **Make sure that the environment variables that the Solaris Management Console uses have the correct values.**

   a. **Set the value of the `DISPLAY` variable.**

   ```
   $ DISPLAY=local:n.n
   $ export DISPLAY=local:n.n
   ```

   b. **Set the value of the `LOGNAME` variable to the user name.**

   ```
   $ LOGNAME=same-username-on-both-systems
   $ export LOGNAME=same-username-on-both-systems
   ```

   c. **Set the value of the `USER` variable to the role name.**

   ```
   $ USER=same-rolename-on-both-systems
   $ export USER=same-rolename-on-both-systems
   ```

5. **In the role, start the Solaris Management Console from the command line.**

   ```
   $ /usr/sbin/smc &
   ```

6 **Select a tool under System Configuration.**

When you select a tool such as User, a dialog box displays the Solaris Management Console server name, your user name, your role name, and a place to type the role's password. Make sure that the entries are correct.

7 **As the role, log in to the server.**

Type the role's password and press Login as Role. You can now use the Solaris Management Console to manage the system.

Note – When you try to access network database information from a system that is not the LDAP server, the operation fails. The Console allows you to log in to the remote host and open the toolbox. However, when you try to access or change information, the following error message indicates that you have selected `Scope=LDAP` on a system that is not the LDAP server:

```
Management server cannot perform the operation requested.
...
Error extracting the value-from-tool.
The keys received from the client were machine, domain, Scope.
Problem with Scope.
```

▼ How to Enable Specific Users to Log In Remotely to the Global Zone in Trusted Extensions

The user's default label range and the zone's default behavior are changed to enable remote login by a non-role. You might want to complete this procedure for a tester who is using a remote labeled system. For security reasons, the tester's system should be running a disjoint label from other users.

Before You Begin You must have a very good reason why this user can log in to the global zone.

You must be in the Security Administrator role in the global zone.

1 **To enable specific users to log in to the global zone, assign them an administrative label range.**

Use the Solaris Management Console to assign a clearance of `ADMIN_HIGH` and a minimum label of `ADMIN_LOW` to each user. For details, see "How to Modify a User's Label Range in the Solaris Management Console" on page 189.

The user's labeled zones must also permit login.

2 **To enable remote login from a labeled zone into the global zone, do the following.**

a. **Add a multilevel port for remote login to the global zone.**

Use the Solaris Management Console. Port 513 over the TCP protocol enables remote login. For an example, see "How to Create a Multilevel Port for a Zone" on page 228.

b. **Read the `tnzonecfg` changes into the kernel.**

```
# tnctl -fz /etc/security/tsol/tnzonecfg
```

c. **Restart the remote login service.**

```
# svcadm restart svc:/network/login:rlogin
```

▼ How to Use Xvnc to Remotely Access a Trusted Extensions System

Virtual Network Computing (vnc) technology connects a client to a remote server, then displays the desktop of the remote server in a window on the client. Xvnc is the UNIX version of vnc, which is based on a standard X server. In Trusted Extensions, a client on any platform can connect to an Xvnc that is running Trusted Extensions software, log in to the Xvnc server, then display and work on a multilevel desktop.

Before You Begin You have installed and configured Trusted Extensions software on the system that is going to be used as the Xvnc server. You have created and booted the labeled zones. Your Xvnc server recognizes the vnc clients by hostname or IP address.

You are superuser in the global zone of the system that is going to be used as the Xvnc server.

1 **Configure the Xvnc server.**

For more information, see the `Xvnc`(1) and `vncconfig`(1) man pages.

Caution – If you are running the Solaris 10 10/08 or the Solaris 10 5/08 release, you must patch your system before configuring the server. For a SPARC system, install the latest version of patch 125719. For an x86 system, install the latest version of patch 125720.

---

a. **Create the Xservers configuration directory.**

```
# mkdir -p /etc/dt/config
```

b. **Copy the `/usr/dt/config/Xservers` file to the `/etc/dt/config` directory.**

```
# cp /usr/dt/config/Xservers /etc/dt/config/Xservers
```

c. **Edit the `/etc/dt/config/Xservers` file to start up the Xvnc program instead of Xserver or Xorg.**

In this example, the entry is configured to log in to the server without a password. To successfully log in the desktop, the local UID must be none instead of `console`.

The entry is split for display purposes. The entry must be on one line.

```
#   :0  Local local_uid@console root /usr/X11/bin/Xserver :0 -nobanner
  :0  Local local_uid@none root /usr/X11/bin/Xvnc :0 -nobanner
  -AlwaysShared -SecurityTypes None -geometry 1024x768x24 -depth 24
```

Note – A safer configuration is to require a password by using the `-SecurityTypes VncAuth` parameter. The `Xvnc`(1) man page describes password requirements.

---

d. **Reboot the server or start the Xvnc server.**

```
# reboot
```

After reboot, verify that the Xvnc program is running.

```
# ps -ef | grep Xvnc
  root  2145  932  0  Jan 18 ?  6:15 /usr/X11/bin/Xvnc :0 -nobanner
  -AlwaysShared -SecurityTypes None -geometry 1024
```

2 **On every vnc client of the Trusted Extensions Xvnc server, install vnc client software.**

For the client system, you have a choice of software. This example uses the Sun vnc software.

```
# cd SUNW-pkg-directory
# pkgadd -d . SUNWvncviewer
```

3 **In a terminal window on a vnc client, connect to the server.**

```
% /usr/bin/vncviewer Xvnc-server-hostname
```

4 **In the window that displays, type your name and password.**

Continue with the login procedure. For a description of the remaining steps, see "Logging In to Trusted Extensions" in *Solaris Trusted Extensions User's Guide*.

If you logged in to the server as superuser, you can administer the server immediately. If you logged in to the server as a user, you must assume a role to administer the system.

CHAPTER 15

Trusted Extensions and LDAP (Overview)

This chapter describes the use of the Sun Java™ System Directory Server (Directory Server) for a system that is configured with Solaris Trusted Extensions.

Using a Naming Service in Trusted Extensions

To achieve uniformity of user, host, and network attributes within a security domain with multiple Trusted Extensions systems, a naming service is used for distributing most configuration information. LDAP is an example of a naming service. The `nsswitch.conf` file determines which naming service is used. LDAP is the recommended naming service for Trusted Extensions.

The Directory Server can provide the LDAP naming service for Trusted Extensions and Solaris clients. The server must include Trusted Extensions network databases, and the Trusted Extensions clients must connect to the server over a multilevel port. The security administrator specifies the multilevel port when configuring Trusted Extensions.

Trusted Extensions adds two trusted network databases to the LDAP server: `tnrhdb` and `tnrhtp`. These databases are administered by using the Security Templates tool in the Solaris Management Console. A toolbox of `Scope=LDAP, Policy=TSOL` stores configuration changes on the Directory Server.

- For information about the use of the LDAP naming service in the Solaris OS, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.
- Setting up the Directory Server for Trusted Extensions clients is described in Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)." Trusted Extensions systems can be clients of a Solaris LDAP server by using an LDAP proxy server that is configured with Trusted Extensions.

Note – Systems that are configured with Trusted Extensions cannot be clients of NIS or NIS+ masters.

Non-Networked Trusted Extensions Systems

If a naming service is not used at a site, administrators must ensure that configuration information for users, hosts, and networks is identical on all hosts. A change that is made on one host must be made on all hosts.

On a non-networked Trusted Extensions system, configuration information is maintained in the `/etc`, `/etc/security`, and `/etc/security/tsol` directories. The Security Templates tool in the Solaris Management Console enables you to modify network database parameters. Users, roles, and rights are modified in the User Accounts, Administrative Roles, and Rights tools. A toolbox on This Computer with `Scope=Files, Policy=TSOL` stores configuration changes locally.

Trusted Extensions LDAP Databases

Trusted Extensions extends the Directory Server's schema to accommodate the `tnrhdb` and `tnrhtp` databases. Trusted Extensions defines two new attributes, `ipTnetNumber` and `ipTnetTemplateName`, and two new object classes, `ipTnetTemplate` and `ipTnetHost`.

The attribute definitions are as follows:

```
ipTnetNumber
   ( 1.3.6.1.1.1.1.34 NAME 'ipTnetNumber'
     DESC 'Trusted network host or subnet address'
     EQUALITY caseExactIA5Match
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
     SINGLE-VALUE )

ipTnetTemplateName
   ( 1.3.6.1.1.1.1.35 NAME 'ipTnetTemplateName'
     DESC 'Trusted network template name'
     EQUALITY caseExactIA5Match
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
     SINGLE-VALUE )
```

The object class definitions are as follows:

```
ipTnetTemplate
   ( 1.3.6.1.1.1.2.18 NAME 'ipTnetTemplate' SUP top STRUCTURAL
     DESC 'Object class for Trusted network host templates'
```

```
     MUST ( ipTnetTemplateName )
     MAY ( SolarisAttrKeyValue ) )

ipTnetHost
   ( 1.3.6.1.1.1.2.19 NAME 'ipTnetHost' SUP top AUXILIARY
     DESC 'Object class for Trusted network host/subnet address
           to template mapping'
     MUST ( ipTnetNumber $ ipTnetTemplateName ) )
```

The `cipso` template definition in LDAP is similar to the following:

```
ou=ipTnet,dc=example,dc=example1,dc=exampleco,dc=com
 objectClass=top
 objectClass=organizationalUnit
 ou=ipTnet

 ipTnetTemplateName=cipso,ou=ipTnet,dc=example,dc=example1,dc=exampleco,dc=com
 objectClass=top
 objectClass=ipTnetTemplate
 ipTnetTemplateName=cipso
 SolarisAttrKeyValue=host_type=cipso;doi=1;min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH;

 ipTnetNumber=0.0.0.0,ou=ipTnet,dc=example,dc=example1,dc=exampleco,dc=com
 objectClass=top
 objectClass=ipTnetTemplate
 objectClass=ipTnetHost
 ipTnetNumber=0.0.0.0
 ipTnetTemplateName=internal
```

Using the LDAP Naming Service in Trusted Extensions

The LDAP naming service is managed in Trusted Extensions as it is managed in the Solaris OS. The following is a sample of useful commands, and contains references to more detailed information:

- For strategies to solve LDAP configuration problems, see Chapter 13, "LDAP Troubleshooting (Reference)," in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.
- To troubleshoot client-to-server LDAP connection problems that are affected by labels, see "How to Debug a Client Connection to the LDAP Server" on page 298.
- To troubleshoot other client-to-server LDAP connection problems, see Chapter 13, "LDAP Troubleshooting (Reference)," in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.
- To display LDAP entries from an LDAP client, type:

```
$ ldaplist -l
$ ldap_cachemgr -g
```

- To display LDAP entries from an LDAP server, type:

```
$ ldap_cachemgr -g
$ idsconfig -v
```

- To list the hosts that LDAP manages, type:

```
$ ldaplist -l hosts       Long listing
$ ldaplist hosts      One-line listing
```

- To list information in the Directory Information Tree (DIT) on LDAP, type:

```
$ ldaplist -l services | more
dn: cn=apocd+ipServiceProtocol=udp,ou=Services,dc=exampleco,dc=com
  objectClass: ipService
  objectClass: top
  cn: apocd
  ipServicePort: 38900
  ipServiceProtocol: udp

...
$ ldaplist services name
dn=cn=name+ipServiceProtocol=udp,ou=Services,dc=exampleco,dc=com
```

- To display the status of the LDAP service on the client, type:

```
# svcs -xv network/ldap/client
svc:/network/ldap/client:default (LDAP client)
 State: online since date
   See: man -M /usr/share/man -s 1M ldap_cachemgr
   See: /var/svc/log/network-ldap-client:default.log
Impact: None.
```

- To start and stop the LDAP client, type:

```
# svcadm enable network/ldap/client
```

```
# svcadm disable network/ldap/client
```

- To start and stop the LDAP server in version 5.2 of Sun Java System Directory Server software, type:

```
# installation-directory/slap-LDAP-server-hostname/start-slapd
# installation-directory/slap-LDAP-server-hostname/stop-slapd
```

- To start and stop the LDAP server in version 6 of Sun Java System Directory Server software, type:

```
# dsadm start /export/home/ds/instances/your-instance
# dsadm stop /export/home/ds/instances/your-instance
```

- To start and stop a proxy LDAP server in version 6 of Sun Java System Directory Server software, type:

```
# dpadm start /export/home/ds/instances/your-instance
# dpadm stop /export/home/ds/instances/your-instance
```

CHAPTER 16

Managing Zones in Trusted Extensions (Tasks)

This chapter describes how non-global zones work on a system that is configured with Solaris Trusted Extensions. Also included are procedures that are unique to zones in Trusted Extensions.

Zones in Trusted Extensions

A properly configured Trusted Extensions system consists of a global zone, which is the operating system instance, and one or more labeled non-global zones. During configuration, Trusted Extensions attaches a unique label to each zone, which creates labeled zones. The labels come from the `label_encodings` file. The administrators can create a zone for each label, but are not required to. It is possible to have more labels than labeled zones on a system. It is not possible to have more labeled zones than labels.

On a Trusted Extensions system, the file systems of a zone are usually mounted as a loopback file system (lofs). All writable files and directories in a labeled zone are at the label of the zone. By default, a user can view files that are in a zone at a lower label than the user's current label. This configuration enables users to view their home directories at lower labels than the label of the current workspace. Although users can view files at a lower label, they cannot modify them. Users can only modify files from a process that has the same label as the file.

In Trusted Extensions, the global zone is an administrative zone. The labeled zones are for regular users. Users can work in a zone whose label is within the user's accreditation range.

Every zone has an associated IP address and security attributes. A zone can be configured with multilevel ports (MLPs). Also, a zone can be configured with a policy for Internet Control Message Protocol (ICMP) broadcasts, such as `ping`.

For information about sharing directories from a labeled zone and about mounting directories from labeled zones remotely, see Chapter 17, "Managing and Mounting Files in Trusted Extensions (Tasks)," and "Mounting Labeled ZFS Datasets" on page 237.

Zones in Trusted Extensions are built on the Solaris zones product. For details, see Part II, "Zones," in *System Administration Guide: Virtualization Using the Solaris Operating System*. In particular, patching and package installation issues affect Trusted Extensions. For details, see Chapter 23, "Packages on an OpenSolaris System With Zones Installed," in *System Administration Guide: Virtualization Using the Solaris Operating System* and Chapter 26, "Troubleshooting Miscellaneous Solaris Zones Problems," in *System Administration Guide: Virtualization Using the Solaris Operating System*.

Zones and IP Addresses in Trusted Extensions

Your initial setup team assigned IP addresses to the global zone and the labeled zones. Three types of configurations are documented in "Creating Labeled Zones" on page 65:

- The system has one IP address for the global zone and all labeled zones.

  This configuration is useful on a system that uses DHCP software to obtain its IP address. If no users are expected to log in, an LDAP server might have this configuration.
- The system has one IP address for the global zone, and one IP address that is shared by all zones, including the global zone. Any zone can have a combination of a unique address and a shared address.

  This configuration is useful on a system that regular users are going to log in to. It can also be used for a printer or an NFS server. This configuration conserves IP addresses.
- The system has one IP address for the global zone, and each labeled zone has a unique IP address.

  This configuration is useful for providing access to separate physical networks of single-level systems. Typically, each zone would have an IP address on a different physical network from the other labeled zones. Because this configuration is implemented with a single IP instance, the global zone controls the physical interfaces and manages global resources, such as the route table.

With the introduction of exclusive IP instances for a non-global zone, a fourth type of configuration is available in the Solaris OS. In the Solaris Express Community Edition, a non-global zone can be assigned its own IP instance and manage its own physical interfaces. In this configuration, each zone operates as if it is a distinct system. For a description, see "Zone Network Interfaces" in *System Administration Guide: Virtualization Using the Solaris Operating System*.

However, in such a configuration, each labeled zone operates as if it is a distinct single-labeled system. The multilevel networking features of Trusted Extensions rely on features of a shared IP stack. Administration procedures in Trusted Extensions assume that networking is controlled entirely by the global zone. Therefore, if your initial setup team has installed labeled zones with exclusive IP instances, you must provide or refer to site-specific documentation.

Zones and Multilevel Ports

By default, a zone cannot send packets to and receive packets from any other zone. Multilevel ports (MLPs) enable particular services on a port to accept requests within a range of labels or from a set of labels. These privileged services can reply at the label of the request. For example, you might want to create a privileged web browser port that can listen at all labels, but whose replies are restricted by label. By default, labeled zones have no MLPs.

The range of labels or set of labels that constrains the packets that the MLP can accept is based on the zone's IP address. The IP address is assigned a remote host template in the `tnrhdb` database. The label range or set of labels in the remote host template constrains the packets that the MLP can accept.

- The constraints on MLPs for different IP address configurations are as follows:
- On a system where the global zone has an IP address and each labeled zone has a unique IP address, an MLP for a particular service can be added to every zone. For example, the system could be configured so that the `ssh` service, over TCP port `22`, is an MLP in the global zone and in every labeled zone.
- In a typical configuration, the global zone is assigned one IP address and labeled zones share a second IP address with the global zone. When an MLP is added to a shared interface, the service packet is routed to the labeled zone where the MLP is defined. The packet is accepted only if the remote host template for the labeled zone includes the label of the packet. If the range is `ADMIN_LOW` to `ADMIN_HIGH`, then all packets are accepted. A narrower range would discard packets that are not within the range.

  At most, one zone can define a particular port to be an MLP on a shared interface. In the preceding scenario, where the `ssh` port is configured as a shared MLP in a non-global zone, no other zone can receive `ssh` connections on the shared address. However, the global zone could define the `ssh` port as a private MLP for receipt of connections on its zone-specific address.
- On a system where the global zone and the labeled zones share an IP address, an MLP for the `ssh` service could be added to one zone. If the MLP for `ssh` is added to the global zone, then no labeled zone can add an MLP for the `ssh` service. Similarly, if the MLP for the `ssh` service is added to a labeled zone, then the global zone cannot be configured with an `ssh` MLP.

For an example of adding MLPs to labeled zones, see Example 19–16.

Zones and ICMP in Trusted Extensions

Networks transmit broadcast messages and send ICMP packets to systems on the network. On a multilevel system, these transmissions could flood the system at every label. By default, the network policy for labeled zones requires that ICMP packets be received only at the matching label.

Global Zone Processes and Labeled Zones

In Trusted Extensions, MAC policy applies to all processes, including processes in the global zone. Processes in the global zone run at the label `ADMIN_HIGH`. When files from a global zone are shared, they are shared at the label `ADMIN_LOW`. Therefore, because MAC prevents a higher-labeled process from modifying a lower-level object, the global zone usually cannot write to an NFS-mounted system.

However, in a limited number of cases, actions in a labeled zone can require that a global zone process modify a file in that zone.

To enable a global zone process to mount a remote file system with read/write permissions, the mount must be under the zone path of the zone whose label corresponds to that of the remote file system. But it must not be mounted under that zone's root path.

- The mounting system must have a zone at the identical label as the remote file system.
- The system must mount the remote file system under the zone path of the identically labeled zone.

  The system must *not* mount the remote file system under the *zone root path* of the identically labeled zone

Consider a zone that is named `public` at the label `PUBLIC`. The *zone path* is `/zone/public/`. All directories under the zone path are at the label `PUBLIC`, as in:

```
/zone/public/dev
/zone/public/etc
/zone/public/home/username
/zone/public/root
/zone/public/usr
```

Of the directories under the zone path, only files under `/zone/public/root` are visible from the public zone. All other directories and files at the label `PUBLIC` are accessible only from the global zone. The path `/zone/public/root` is the *zone root path*.

From the perspective of the public zone administrator, the zone root path is visible as `/`. Similarly, the public zone administrator cannot access a user's home directory in the zone path, `/zone/public/home/`*username* directory. That directory is visible only from the global zone. The public zone mounts that directory in the zone root path as `/home/`*username*. From the perspective of the global zone, that mount is visible as `/zone/public/root/home/`*username*.

The public zone administrator can modify `/home/`*username*. A global zone process, when files in a user's home directory need to be modified, does not use that path. The global zone uses the user's home directory in the zone path, `/zone/public/home/`*username*.

- Files and directories that are under the zone path, `/zone/`*zonename*`/`, but not under the zone root path, `/zone/`*zonename*`/root` directory, can be modified by a global zone process that runs at the label `ADMIN_HIGH`.
- Files and directories that are under the zone root path, `/zone/public/root`, can be modified by the labeled zone administrator.

For example, when a user allocates a device in the public zone, a global zone process that runs at the label `ADMIN_HIGH` modifies the `dev` directory in the zone path, `/zone/public/dev`. Similarly, when a user saves a desktop configuration, the desktop configuration file is modified by a global zone process in the `/zone/public/home/`*username*. Finally, to share files from a labeled zone, the global zone administrator creates the configuration file, `dfstab`, in the zone path, `/zone/public/etc/dfs/dfstab`. A labeled zone administrator cannot access that file, and cannot share files from the labeled zone. To share a labeled directory, see "How to Share Directories From a Labeled Zone" on page 239.

Zone Administration Utilities in Trusted Extensions

Some zone administration tasks can be performed from the command line. However, the simplest way to administer zones is to use the GUIs that Trusted Extensions provides:

- The configuration of zone security attributes is performed by using the Trusted Network Zones tool in the Solaris Management Console. For a description of the tool, see "Trusted Network Zones Tool" on page 144. For examples of zone configuration and creation, see Chapter 4, "Configuring Trusted Extensions (Tasks)," and "How to Create a Multilevel Port for a Zone" on page 228.
- The shell script, `/usr/sbin/txzonemgr`, provides a menu-based wizard for creating, installing, initializing, and booting zones. `txzonemgr` uses the `zenity` command. For details, see the `zenity`(1) man page.

Managing Zones (Task Map)

The following task map describes zone management tasks that are specific to Trusted Extensions. The map also points to common procedures that are performed in Trusted Extensions just as they are performed on a Solaris system.

Task	Description	For Instructions
View all zones.	At any label, views the zones that are dominated by the current zone.	"How to Display Ready or Running Zones" on page 218

Task	Description	For Instructions
View mounted directories.	At any label, views the directories that are dominated by the current label.	"How to Display the Labels of Mounted Files" on page 220
Enable regular users to view an `/etc` file.	Loopback mounts a directory or file from the global zone that is not visible by default in a labeled zone.	"How to Loopback Mount a File That Is Usually Not Visible in a Labeled Zone" on page 221
Prevent regular users from viewing a lower-level home directory from a higher label.	By default, lower-level directories are visible from higher-level zones. When you disable the mounting of one lower-level zone, you disable all mounts of lower-level zones.	"How to Disable the Mounting of Lower-Level Files" on page 222
Configure a zone to enable the changing of the labels on files.	Labeled zones have limited privileges. By default, labeled zones do not have the privilege that enables an authorized user to relabel a file. You modify the zone configuration to add the privilege.	"How to Enable Files to be Relabeled From a Labeled Zone" on page 226
Attach a ZFS dataset to a labeled zone and share it.	Mounts a ZFS dataset with read/write permissions in a labeled zone and shares the dataset read-only with a higher zone.	"How to Share a ZFS Dataset From a Labeled Zone" on page 224.
Configure a new zone.	Creates a zone at a label that is not currently being used to label a zone on this system.	See "Create the First Labeled Zone" on page 66. Then, follow the procedure that the initial setup team used to create the other zones. For the steps, see "Creating Labeled Zones" on page 65.
Create a multilevel port for an application.	Multilevel ports are useful for programs that require a multilevel feed into a labeled zone.	"How to Configure a Multilevel Port for NFSv3 Over `udp`" on page 227 "How to Create a Multilevel Port for a Zone" on page 228
Troubleshoot NFS mount and access problems.	Debugs general access issues for mounts and possibly for zones.	"How to Troubleshoot Mount Failures in Trusted Extensions" on page 246
Remove a labeled zone.	Completely removes a labeled zone from the system.	"How to Remove a Non-Global Zone" in *System Administration Guide: Virtualization Using the Solaris Operating System*

▼ How to Display Ready or Running Zones

This procedure creates a shell script that displays the labels of the current zone and all zones that the current zone dominates.

Before You Begin You must be in the System Administrator role in the global zone.

1. **Use the trusted editor to create the `getzonelabels` script.**

   For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

   Provide the pathname to the script, such as `/usr/local/scripts/getzonelabels`.

2. **Add the following content, and save the file:**

```
#!/bin/sh
#
echo "NAME\t\tSTATUS\t\tLABEL"
echo "====\t\t======\t\t====="
myzone=‘zonename‘
for i in ‘/usr/sbin/zoneadm list -p‘ ; do
        zone=‘echo $i | cut -d ":" -f2‘
        status=‘echo $i | cut -d ":" -f3‘
        path=‘echo $i | cut -d ":" -f4‘
        if [ $zone != global ]; then
                if [ $myzone = global ]; then
                        path=$path/root/tmp
                else
                        path=$path/export/home
                fi
        fi
        label=‘/usr/bin/getlabel -s $path |cut -d ":" -f2-9‘
        if [ ‘echo $zone|wc -m‘ -lt 8 ]; then
                echo "$zone\t\t$status\t$label"
        else
                echo "$zone\t$status\t$label"
        fi
done
```

3. **Test the script in the global zone.**

```
# getzonelabels
NAME            STATUS          LABEL
====            ======          =====
global          running         ADMIN_HIGH
needtoknow      running         CONFIDENTIAL : NEED TO KNOW
restricted      ready           CONFIDENTIAL : RESTRICTED
internal        running         CONFIDENTIAL : INTERNAL
public          running         PUBLIC
```

   When run from the global zone, the script displays the labels of all ready or running zones. Here is the global zone output for the zones that were created from the default `label_encodings` file:

Example 16–1 Displaying the Labels of All Ready or Running Zones

In the following example, a user runs the `getzonelabels` script in the `internal` zone.

```
# getzonelabels
NAME            STATUS          LABEL
====            ======          =====
internal        running         CONFIDENTIAL : INTERNAL
public          running         PUBLIC
```

How to Display the Labels of Mounted Files

This procedure creates a shell script that displays the mounted file systems of the current zone. When run from the global zone, the script displays the labels of all mounted file systems in every zone.

Before You Begin You must be in the System Administrator role in the global zone.

1. **Use the trusted editor to create the `getmounts` script.**

   For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

   Provide the pathname to the script, such as `/usr/local/scripts/getmounts`.

2. **Add the following content and save the file:**

```
#!/bin/sh
#
for i in ‘/usr/sbin/mount -p | cut -d " " -f3‘ ; do
        /usr/bin/getlabel $i
done
```

3. **Test the script in the global zone.**

```
# /usr/local/scripts/getmounts
/:      ADMIN_LOW
/dev:   ADMIN_LOW
/kernel:        ADMIN_LOW
/lib:   ADMIN_LOW
/opt:   ADMIN_LOW
/platform:      ADMIN_LOW
/sbin:  ADMIN_LOW
/usr:   ADMIN_LOW
/var/tsol/doors:        ADMIN_LOW
/zone/needtoknow/export/home:   CONFIDENTIAL : NEED TO KNOW
/zone/internal/export/home:     CONFIDENTIAL : INTERNAL USE ONLY
/zone/restricted/export/home:   CONFIDENTIAL : RESTRICTED
/proc:  ADMIN_LOW
```

```
/system/contract:       ADMIN_LOW
/etc/svc/volatile:      ADMIN_LOW
/etc/mnttab:    ADMIN_LOW
/dev/fd:        ADMIN_LOW
/tmp:           ADMIN_LOW
/var/run:       ADMIN_LOW
/zone/public/export/home:  PUBLIC
/root:          ADMIN_LOW
```

Example 16–2 Displaying the Labels of File Systems in the `restricted` Zone

When run from a labeled zone by a regular user, the `getmounts` script displays the labels of all the mounted file systems in that zone. On a system where zones are created for every label in the default `label_encodings` file, the following is the output from the `restricted` zone:

```
# /usr/local/scripts/getmounts
/:      CONFIDENTIAL : RESTRICTED
/dev:   CONFIDENTIAL : RESTRICTED
/kernel:        ADMIN_LOW
/lib:   ADMIN_LOW
/opt:   ADMIN_LOW
/platform:      ADMIN_LOW
/sbin:  ADMIN_LOW
/usr:   ADMIN_LOW
/var/tsol/doors:        ADMIN_LOW
/zone/needtoknow/export/home:   CONFIDENTIAL : NEED TO KNOW
/zone/internal/export/home:     CONFIDENTIAL : INTERNAL USE ONLY
/proc:  CONFIDENTIAL : RESTRICTED
/system/contract:       CONFIDENTIAL : RESTRICTED
/etc/svc/volatile:      CONFIDENTIAL : RESTRICTED
/etc/mnttab:    CONFIDENTIAL : RESTRICTED
/dev/fd:        CONFIDENTIAL : RESTRICTED
/tmp:   CONFIDENTIAL : RESTRICTED
/var/run:       CONFIDENTIAL : RESTRICTED
/zone/public/export/home:       PUBLIC
/home/gfaden:   CONFIDENTIAL : RESTRICTED
```

▼ How to Loopback Mount a File That Is Usually Not Visible in a Labeled Zone

This procedure enables a user in a specified labeled zone to view files that are not exported from the global zone by default.

Before You Begin You must be in the System Administrator role in the global zone.

1. **Halt the zone whose configuration you want to change.**

```
# zoneadm -z zone-name halt
```

2. **Loopback mount a file or directory.**

For example, enable ordinary users to view a file in the `/etc` directory.

```
# zonecfg -z zone-name
 add filesystem
 set special=/etc/filename
 set directory=/etc/filename
 set type=lofs
 add options [ro,nodevices,nosetuid]
 end
 exit
```

Note – Certain files are not used by the system, so that loopback mounting them has no effect. For example, the `/etc/dfs/dfstab` file in a labeled zone is not checked by Trusted Extensions software. For more information, see "Sharing Files From a Labeled Zone" on page 233.

---

3. **Start the zone.**

```
# zoneadm -z zone-name boot
```

Example 16–3 Loopback Mounting the `/etc/passwd` file

In this example, the security administrator wants to enable testers and programmers to check that their local passwords are set. After the `sandbox` zone is halted, it is configured to loopback mount the `passwd` file. Then, the zone is restarted.

```
# zoneadm -z sandbox halt
# zonecfg -z sandbox
 add filesystem
    set special=/etc/passwd
    set directory=/etc/passwd
    set type=lofs
    add options [ro,nodevices,nosetuid]
 end
 exit
# zoneadm -z sandbox boot
```

▼ How to Disable the Mounting of Lower-Level Files

By default, users can view lower-level files. Remove the `net_mac_aware` privilege to prevent the viewing of all lower-level files from a particular zone. For a description of the `net_mac_aware` privilege, see the `privileges(5)` man page.

Before You Begin You must be in the System Administrator role in the global zone.

1. **Halt the zone whose configuration you want to change.**

```
# zoneadm -z zone-name halt
```

2. **Configure the zone to prevent the viewing of lower-level files.**

   Remove the `net_mac_aware` privilege from the zone.

```
# zonecfg -z zone-name
 set limitpriv=default,!net_mac_aware
 exit
```

3. **Restart the zone.**

```
# zoneadm -z zone-name boot
```

Example 16–4 Preventing Users From Viewing Lower-Level Files

In this example, the security administrator wants to prevent users on one system from being confused. Therefore, users can only view files at the label at which the users are working. So, the security administrator prevents the viewing of all lower-level files. On this system, users cannot see publicly available files unless they are working at the `PUBLIC` label. Also, users can only NFS mount files at the label of the zones.

```
# zoneadm -z restricted halt
# zonecfg -z restricted
 set limitpriv=default,!net_mac_aware
 exit
# zoneadm -z restricted boot

# zoneadm -z needtoknow halt
# zonecfg -z needtoknow
 set limitpriv=default,!net_mac_aware
 exit
# zoneadm -z needtoknow boot

# zoneadm -z internal halt
# zonecfg -z internal
 set limitpriv=default,!net_mac_aware
 exit
# zoneadm -z internal boot
```

Because `PUBLIC` is the lowest label, the security administrator does not run the commands for the `PUBLIC` zone.

How to Share a ZFS Dataset From a Labeled Zone

In this procedure, you mount a ZFS dataset with read/write permissions in a labeled zone. Because all commands are executed in the global zone, the global zone administrator controls the addition of ZFS datasets to labeled zones.

At a minimum, the labeled zone must be in the `ready` state to share a dataset. The zone can be in the `running` state.

Before You Begin To configure the zone with the dataset, you first halt the zone.

1. **Create the ZFS dataset.**

   ```
   # zfs create datasetdir/subdir
   ```

   The name of the dataset can include a directory, such as `zone/data`.

2. **In the global zone, halt the labeled zone.**

   ```
   # zoneadm -z labeled-zone-name halt
   ```

3. **Set the mount point of the dataset.**

   ```
   # zfs set mountpoint=legacy datasetdir/subdir
   ```

   Setting the ZFS `mountpoint` property sets the label of the mount point when the mount point corresponds to a labeled zone.

4. **Add the dataset to the zone as a file system.**

   ```
   # zonecfg -z labeled-zone-name
   # zonecfg:labeled-zone-name> add fs
   # zonecfg:labeled-zone-name:dataset> set dir=/subdir
   # zonecfg:labeled-zone-name:dataset> set special=datasetdir/subdir
   # zonecfg:labeled-zone-name:dataset> set type=zfs
   # zonecfg:labeled-zone-name:dataset> end
   # zonecfg:labeled-zone-name> exit
   ```

   By adding the dataset as a file system, the dataset is mounted at `/data` in the zone before the `dfstab` file is interpreted. This step ensures that the dataset is not mounted before the zone is booted. Specifically, the zone boots, the dataset is mounted, then the `dfstab` file is interpreted.

5. **Share the dataset.**

   Add an entry for the dataset file system to the `/zone/`*labeled-zone-name*`/etc/dfs/dfstab` file. This entry also uses the `/`*subdir* pathname.

   ```
   share  -F nfs  -d "dataset-comment"  /subdir
   ```

6. **Boot the labeled zone.**

   ```
   # zoneadm -z labeled-zone-name boot
   ```

When the zone is booted, the dataset is mounted automatically as a read/write mount point in the *labeled-zone-name* zone with the label of the *labeled-zone-name* zone.

Example 16–5 Sharing and Mounting a ZFS Dataset From Labeled Zones

In this example, the administrator adds a ZFS dataset to the `needtoknow` zone and shares the dataset. The dataset, `zone/data`, is currently assigned to the `/mnt` mount point. Users in the `restricted` zone can view the dataset.

First, the administrator halts the zone.

```
# zoneadm -z needtoknow halt
```

Because the dataset is currently assigned to a different mount point, the administrator removes the previous assignment, then sets the new mount point.

```
# zfs set zoned=off zone/data
# zfs set mountpoint=legacy zone/data
```

Next, in the `zonecfg` interactive interface, the administrator explicitly adds the dataset to the `needtoknow` zone.

```
# zonecfg -z needtoknow
# zonecfg:needtoknow> add fs
# zonecfg:needtoknow:dataset> set dir=/data
# zonecfg:needtoknow:dataset> set special=zone/data
# zonecfg:needtoknow:dataset> set type=zfs
# zonecfg:needtoknow:dataset> end
# zonecfg:needtoknow> exit
```

Next, the administrator modifies the `/zone/needtoknow/etc/dfs/dfstab` file to share the dataset, then boots the `needtoknow` zone.

```
## Global zone dfstab file for needtoknow zone
share  -F nfs  -d "App Data on ZFS"  /data

# zoneadm -z needtoknow boot
```

The dataset is now accessible.

Users in the the `restricted` zone, which dominates the `needtoknow` zone, can view the mounted dataset by changing to the `/data` directory. They use the full path to the mounted dataset from the perspective of the global zone. In this example, `machine1` is the host name of the system that includes the labeled zone. The administrator assigned this host name to a non-shared IP address.

```
# cd /net/machine1/zone/needtoknow/root/data
```

Troubleshooting If the attempt to reach the dataset from the higher label returns the error `not found` or `No such file or directory`, the administrator must restart the automounter service by running the `svcadm restart autofs` command.

▼ How to Enable Files to be Relabeled From a Labeled Zone

This procedure is a prerequisite for a user to be able to relabel files.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Halt the zone whose configuration you want to change.**

```
# zoneadm -z zone-name halt
```

2. **Configure the zone to enable relabeling.**

   Add the appropriate privileges to the zone. The windows privileges enable users to use drag-and-drop and cut-and-paste operations.

   - **To enable downgrades, add the `file_downgrade_sl` privilege to the zone.**

```
# zonecfg -z zone-name
 set limitpriv=default,win_dac_read,win_mac_read,win_dac_write,
 win_mac_write,win_selection,file_downgrade_sl
 exit
```

   - **To enable upgrades, add the `sys_trans_label` and `file_upgrade_sl` privileges to the zone.**

```
# zonecfg -z zone-name
 set limitpriv=default,win_dac_read,win_mac_read,win_dac_write,
 win_mac_write,win_selection,sys_trans_label,file_upgrade_sl
 exit
```

   - **To enable both upgrades and downgrades, add all three privileges to the zone.**

```
# zonecfg -z zone-name
 set limitpriv=default,win_dac_read,win_mac_read,win_dac_write,
 win_mac_write,win_selection,sys_trans_label,file_downgrade_sl,
 file_upgrade_sl
 exit
```

3. **Restart the zone.**

```
# zoneadm -z zone-name boot
```

For the user and process requirements that permit relabeling, see the setflabel(3TSOL) man page. To authorize a user to relabel files, see "How to Enable a User to Change the Security Level of Data" on page 195.

Example 16–6 Enabling Upgrades From the `internal` Zone

In this example, the security administrator wants to enable authorized users on a system to upgrade files. By enabling users to upgrade information, the administrator enables them to protect the information at a higher level of security. In the global zone, the administrator runs the following zone administration commands.

```
# zoneadm -z internal halt
# zonecfg -z internal
 set limitpriv=default,sys_trans_label,file_upgrade_sl
 exit
# zoneadm -z internal boot
```

Authorized users can now upgrade `internal` information to `restricted` from the `internal` zone.

Example 16–7 Enabling Downgrades From the `restricted` Zone

In this example, the security administrator wants to enable authorized users on a system to downgrade files. Because the administrator does not add windows privileges to the zone, authorized users cannot use the File Manager to relabel files. To relabel files, users use the `setlabel` command.

By enabling users to downgrade information, the administrator permits users at a lower level of security to access the files. In the global zone, the administrator runs the following zone administration commands.

```
# zoneadm -z restricted halt
# zonecfg -z restricted
 set limitpriv=default,file_downgrade_sl
 exit
# zoneadm -z restricted boot
```

Authorized users can now downgrade `restricted` information to `internal` or `public` from the `restricted` zone by using the `setlabel` command.

▼ How to Configure a Multilevel Port for NFSv3 Over `udp`

This procedure is used to enable NFSv3 read-down mounts over `udp`. The Solaris Management Console is used to add the MLP.

Before You Begin You must be in the Security Administrator role in the global zone.

1 **Start the Solaris Management Console.**

For details, see "How to Administer the Local System With the Solaris Management Console" on page 154.

2 **Choose the Files toolbox.**

The title of the toolbox includes `Scope=Files, Policy=TSOL`.

3 **Configure the zone and the MLP.**

a. **Navigate to the Trusted Network Zones tool.**

b. **Double-click the global zone.**

c. **Add a multilevel port for the UDP protocol:**

i. **Click Add for the Multilevel Ports for Zone's IP Addresses.**

ii. **Type 2049 for the port number, and click OK.**

d. **Click OK to save the settings.**

4 **Close the Solaris Management Console.**

5 **Update the kernel.**

```
# tnctl -fz /etc/security/tsol/tnzonecfg
```

How to Create a Multilevel Port for a Zone

This procedure is used when an application that runs in a labeled zone requires a multilevel port (MLP) to communicate with the zone. In this procedure, a web proxy communicates with the zone. The Solaris Management Console is used to add the MLP.

Before You Begin You must be in the Security Administrator role in the global zone. The labeled zone must exist. For details, see "Creating Labeled Zones" on page 65.

1 **Start the Solaris Management Console.**

For details, see "How to Administer the Local System With the Solaris Management Console" on page 154.

2 **Choose the Files toolbox.**

The title of the toolbox includes `Scope=Files, Policy=TSOL`.

3 **Add the proxy host and the webservices host to the list of computers.**

a. **Under System Configuration, navigate to the Computers and Networks tool.**

b. **In the Computers tool, click the Action menu and choose Add Computer.**

c. **Add the host name and IP address for the proxy host.**

d. **Save the changes.**

e. **Add the host name and IP address for the webservice host.**

f. **Save the changes.**

4 **Configure the zone and the MLP.**

a. **Navigate to the Trusted Network Zones tool.**

b. **Select the labeled zone.**

c. **In the MLP Configuration for Local IP Addresses section, specify the appropriate port/protocol field.**

d. **Save the changes.**

5 **For the zone, customize a template by completing the following steps:**

a. **Navigate to the Security Templates tool.**

   Click the Action menu and choose Add Template.

b. **Use the host name for the template name.**

c. **Specify CIPSO for the Host Type.**

d. **Use the label of the zone for the Minimum Label and for the Maximum Label.**

e. **Assign the zone label to the Security Label Set.**

f. **Select the Hosts Explicitly Assigned tab.**

g. **In the Add an Entry section, add the IP address that is associated with the zone.**

h. **Save the changes.**

6 **Close the Solaris Management Console.**

7 **Start the zones.**

```
# zoneadm -z zone-name boot
```

8 **In the global zone, add routes for the new addresses.**

For example, if the zones have a shared IP address, do the following:

```
# route add proxy labeled-zones-IP-address
# route add webservice labeled-zones-IP-address
```

CHAPTER 17

Managing and Mounting Files in Trusted Extensions (Tasks)

This chapter describes how LOFS, NFS, and ZFS mounts work on a system that is configured with Trusted Extensions. This chapter also covers how to back up and restore files.

Sharing and Mounting Files in Trusted Extensions

Trusted Extensions software supports the same file systems and file system management commands as the Solaris OS. Trusted Extensions adds the ability for a non-global zone to share files. In addition, Trusted Extensions attaches a unique label to every non-global zone. All the files and directories that belong to that zone are mounted at the label of the zone. Any shared file systems that belong to other zones or to NFS servers are mounted at the label of the owner. Trusted Extensions prevents any mounts that would violate the mandatory access control (MAC) policies for labeling. For example, a zone's label must dominate all of its mounted file system labels, and only equally labeled file systems can be mounted with read/write permissions.

NFS Mounts in Trusted Extensions

NFS mounts in Trusted Extensions are similar to Solaris mounts. The differences occur in the use of zone root pathnames when mounting a labeled zone in Trusted Extensions, and in the enforcement of MAC policy.

NFS shares in Trusted Extensions are similar to Solaris shares in a global zone. However, the sharing of files from a labeled zone on a multilevel system is unique to Trusted Extensions:

- **Shares and mounts in the global zone –** Sharing and mounting files in the global zone of a Trusted Extensions system is almost identical to the procedure in the Solaris OS. For mounting files, the automounter, the `vfstab` file, and the `mount` command can be used. For sharing files, the `dfstab` file is used.
- **Mounts in labeled zones –** Mounting files in labeled zones in Trusted Extensions is almost identical to mounting files in non-global zones in the Solaris OS. For mounting files, the automounter, the `vfstab` file, and the `mount` command can be used. In Trusted Extensions, a unique `automount_home_`*label* configuration file exists for each labeled zone.
- **Shares in labeled zones –** Files in a labeled zone can be shared at the label of the zone by using a `dfstab` file that is at the label of the zone, but is visible to the global zone only. So, configuring a labeled zone to share files is performed by the global zone administrator in the global zone. This configuration file is not visible from its labeled zone. For more discussion, see “Global Zone Processes and Labeled Zones” on page 216.

Labels affect which files can be mounted. Files are shared and mounted at a particular label. For a Trusted Extensions client to write to a file that is NFS-mounted, the file must be mounted with read/write permissions *and* be at the same label as the client. If you are mounting a file between two Trusted Extensions hosts, the server and the client must have compatible remote host templates of type `cipso`. If you are mounting a file between a Trusted Extensions host and an unlabeled host, files that are at the single label that is specified for the unlabeled host in the `tnrhdb` file can be mounted. Files that are mounted with LOFS can be viewed, but cannot be modified. For details on NFS mounts, see “Access to NFS Mounted Directories in Trusted Extensions” on page 234.

Labels also affect which directories and files can be viewed. By default, lower-level objects are available in a user's environment. Therefore, in the default configuration, a regular user can view files that are in a zone at a lower level than the user's current level. For example, users can see their lower-level home directories from a higher label. For details, see “Home Directory Creation in Trusted Extensions” on page 234.

If site security forbids the viewing of lower-level objects, you can make lower-level directories invisible to the user. For details, see “How to Disable the Mounting of Lower-Level Files” on page 222.

The mount policy in Trusted Extensions has no MAC overrides. Mounted files that are visible at a lower label can never be modified by a higher-label process. This MAC policy is also in effect in the global zone. A global zone `ADMIN_HIGH` process cannot modify an NFS-mounted file at a lower label, such as a `PUBLIC` file or an `ADMIN_LOW` file. MAC policies enforce the default configuration and are invisible to regular users. Regular users cannot see objects unless they have MAC access to them.

Sharing Files From a Labeled Zone

In the Solaris OS, a non-global zone cannot share directories from its zone. However, in Trusted Extensions, a labeled zone can share directories. The specification of which directories in a labeled zone can be shared is performed in the global zone by using a directory that is outside the `root` path of the zone. For more discussion, see "Global Zone Processes and Labeled Zones" on page 216.

`/zone/`*labeled-zone*`/`*directories* — Also called the zone path. Is the path from the global zone to the labeled zone. Every directory under *labeled-zone* is labeled the same as the zone.

`/zone/`*labeled-zone*`/root/`*directories* — Also called the zone root path. Is the `root` path of a labeled zone from the perspective of the global zone. From the perspective of the labeled zone, this is the zone's root, the `/` directory. This path is not used by the global zone to administer the zone.

To share directories from a labeled zone, the global zone administrator creates and modifies the `dfstab` file in the `/etc` directory of the zone path:

```
/zone/labeled-zone/etc/dfs/dfstab
```

This `/etc` directory is not visible from the labeled zone. This directory is distinct from the `/etc` directory that is visible from the zone:

```
Global zone view: /zone/labeled-zone/root/etc
Labeled zone view of the same directory: /etc
```

A `dfstab` file in this path does not enable labeled directories to be shared.

When the status of the labeled zone is `ready` or `running`, the files that are listed in the `/zone/`*labeled-zone*`/etc/dfs/dfstab` file are shared at the label of the zone. For the procedure, see "How to Share Directories From a Labeled Zone" on page 239.

Access to NFS Mounted Directories in Trusted Extensions

By default, NFS-mounted file systems are visible at the label of the exported file system. If the file system is exported with read/write permissions, users at that label can write to the files. NFS mounts that are at a lower label than the user's current session are visible to the user, but cannot be written to. Even if a file system is shared with read/write permissions, the mounting system can write to it only at the label of the mount.

To make lower-level directories that are NFS-mounted visible to users in a higher-level zone, the administrator of the global zone on the NFS server must export the parent directory. The parent directory is exported at its label. On the client side, each zone must have the `net_mac_aware` privilege. By default, labeled zones include the `net_mac_aware` privilege in their `limitpriv` set.

- **Server configuration –** On the NFS server, you export the parent directory in a `dfstab` file. If the parent directory is in a labeled zone, the `dfstab` file must be modified in the labeled zone of the parent directory. The `dfstab` file for a labeled zone is visible only from the global zone. For the procedure, see "How to Share Directories From a Labeled Zone" on page 239.
- **Client configuration –** The `net_mac_aware` privilege must be specified in the zone configuration file that is used during initial zone configuration. So, a user who is permitted to view all lower-level home directories must have the `net_mac_aware` privilege in every zone, except the lowest zone. For an example, see "How to NFS Mount Files in a Labeled Zone" on page 241.

EXAMPLE 17–1 Providing Access to Lower-Level Home Directories

On the home directory server, the administrator creates and modifies the `/zone/`*labeled-zone*`/etc/dfs/dfstab` file in every labeled zone. The `dfstab` file exports the `/export/home` directory with read/write permissions. Thus, when the directory is mounted at the same label, the home directory is writable. To export the `/export/home` directory of `PUBLIC`, the administrator creates a workspace at the `PUBLIC` label on the home directory server, and from the global zone, modifies the `/zone/public/etc/dfs/dfstab` file.

On the client, the administrator of the global zone checks that every labeled zone, except the lowest label, has the `net_mac_aware` privilege. This privilege permits the mount. This privilege can be specified by using the `zonecfg` command during zone configuration. The lower-level home directory can only be viewed. MAC protects the files in the directory from modification.

Home Directory Creation in Trusted Extensions

Home directories are a special case in Trusted Extensions. You need to make sure that the home directories are created in every zone that a user can use. Also, the home directory mount points must be created in the zones on the user's system. For NFS-mounted home directories to work correctly, the conventional location for directories, `/export/home`, must be used. In Trusted Extensions, the automounter has been modified to handle home directories in every zone, that is, at every label. For details, see "Changes to the Automounter in Trusted Extensions" on page 235.

Home directories are created when users are created. In Trusted Extensions, the Solaris Management Console (Console) is used to create users, so the Console creates the home directories. However, the Console creates the home directories in the global zone of the home directory server. On that server, the directories are mounted by LOFS. Home directories are automatically created by the automounter if they are specified as LOFS mounts.

Note – When you delete a user by using the Console, only the user's home directory in the global zone is deleted. The user's home directories in the labeled zones are not deleted. You are responsible for archiving and deleting the home directories in the labeled zones. For the procedure, see "How to Delete a User Account From a Trusted Extensions System" on page 196.

However, the automounter cannot automatically create home directories on remote NFS servers. Either the user must first log in to the NFS server or administrative intervention is required. To create home directories for users, see "Enable Users to Access Their Home Directories in Trusted Extensions" on page 91.

Changes to the Automounter in Trusted Extensions

In Trusted Extensions, each label requires a separate home directory mount. The `automount` command has been modified to handle these labeled automounts. For each zone, the automounter, `autofs`, mounts an `auto_home_`*zone-name* file. For example, the following is the entry for the global zone in the `auto_home_global` file:

```
+auto_home_global
*       -fstype=lofs    :/export/home/&
```

When a zone that permits lower-level zones to be mounted is booted, the following occurs. The home directories of lower-level zones are mounted read only under `/zone/<`*zone-name*`>/export/home`. The `auto_home_<`*zone-name*`>` map specifies the `/zone` path as the source directory for an `lofs` remount onto `/zone/<`*zone-name*`>/home/<`*username*`>`.

For example, the following is an `auto_home_public` entry in an `auto_home_`*zone-at-higher-label* map that is generated from a higher-level zone:

```
+auto_home_public
*       -fstype=lofs    :/zone/public/export/home/&
```

The following is the corresponding entry in the public zone:

```
auto_home_public
*       -fstype=lofs    :/export/home/&
```

When a home directory is referenced and the name does not match any entries in the `auto_home_`<*zone-name*> map, the map tries to match this loopback mount specification. The software creates the home directory when the following two conditions are met:

1. The map finds the match of the loopback mount specification
2. The home directory name matches a valid user whose home directory does not yet exist in *zone-name*

For details on changes to the automounter, see the `automount(1M)` man page.

Trusted Extensions Software and NFS Protocol Versions

In the Solaris Express Community Edition, Trusted Extensions software recognizes labels on NFS Version 3 (NFSv3) and NFSv4. You can use one of the following sets of mount options:

```
vers=4 proto=tcp
vers=3 proto=tcp
vers=3 proto=udp
```

Trusted Extensions has no restrictions on mounts over the `tcp` protocol. In NFSv3 and NFSv4, the `tcp` protocol can be used for same-label mounts and for read-down mounts. Read-down mounts require a multilevel port (MLP).

For NFSv3, Trusted Extensions behaves like the Solaris OS. The `udp` protocol is the default for NFSv3, but `udp` is used only for the initial mount operation. For subsequent NFS operations, the system uses `tcp`. Therefore, read-down mounts work for NFSv3 in the default configuration.

In the rare case that you have restricted NFSv3 mounts to use the udp protocol for initial and subsequent NFS operations, you must create an MLP for NFS operations that use the `udp` protocol. For the procedure, see "How to Configure a Multilevel Port for NFSv3 Over `udp`" on page 227.

A host that is configured with Trusted Extensions can also share its own file systems with unlabeled hosts. A file or directory that is exported to an unlabeled host is *writable* if its label equals the label that is associated with the remote host in its trusted networking database entries. A file or directory that is exported to an unlabeled host is *readable* only if its label is dominated by the label that is associated with the remote host.

Communications with systems that are running a release of Trusted Solaris software is possible only at a single label. The Trusted Extensions system and the Trusted Solaris system must assign to the other system a template with the unlabeled host type. The unlabeled host types must specify the same single label. As an unlabeled NFS client of a Trusted Solaris server, the label of the client cannot be `ADMIN_LOW`.

The NFS protocol that is used is independent of the local file system's type. Rather, the protocol depends on the type of the sharing computer's operating system. The file system type that is specified to the `mount` command or in the `vfstab` file for remote file systems is always NFS.

Mounting Labeled ZFS Datasets

You can apply a label to a ZFS dataset or mount a ZFS dataset with no label to a zone. The initially unlabeled ZFS dataset acquires the label of the mounting zone.

ZFS provides a security label attribute, `mlslabel`, that contains the label of the data in the dataset. The `mlslabel` property is inheritable. If the property is undefined, it defaults to the string `none`, which indicates no label.

When you mount a ZFS dataset in a labeled zone, the following occurs:

- If the dataset is not labeled, the value of the `mlslabel` property is changed to the label of the mounting zone.

  For the global zone, the `mlslabel` property is not set automatically. If you explicitly label the dataset `admin_low`, the dataset must be mounted read-only.
- If the dataset is labeled, the kernel verifies that the dataset label matches the label of the mounting zone. If the labels do not match, the mount fails.

  If read-down mounts are allowed in the zone, a lower-level dataset mounts read-only.

To set the `mlslabel` property from the command line, type something similar to the following:

```
# zfs set mlslabel=public export/publicinfo
```

The `file_upgrade_sl` privilege is required to set an initial label or to change a non-default label to a higher-level label. The `file_downgrade_sl` privilege is required to remove a label, that is, to set the label to `none`. This privilege is also required to change a non-default label to a lower-level label. When a ZFS dataset has an explicit label, the dataset cannot be mounted on a Solaris system that is not configured with Trusted Extensions.

Backing Up, Sharing, and Mounting Labeled Files (Task Map)

The following task map describes common tasks that are used to back up and restore data from labeled file systems, and to share and mount directories and files that are labeled.

Task	Description	For Instructions
Back up files.	Protects your data by backing it up.	"How to Back Up Files in Trusted Extensions" on page 238
Restore data.	Restores data from a backup.	"How to Restore Files in Trusted Extensions" on page 239
Share the contents of a directory from a labeled zone.	Allows the contents of a labeled directory to be shared among users.	"How to Share Directories From a Labeled Zone" on page 239
Mount the contents of a directory that was shared by a labeled zone.	Allows the contents of a directory to be mounted in a zone at the same label for read/write. When a higher-level zone mounts the shared directory, the directory is mounted read-only.	"How to NFS Mount Files in a Labeled Zone" on page 241
Create home directory mount points.	Creates mount points for every user at every label. This task enables users to access their home directory on a system that is not the NFS home directory server.	"Enable Users to Access Their Home Directories in Trusted Extensions" on page 91
Hide lower-level information from a user who is working at a higher label.	Prevent the viewing of lower-level information from a higher-level window.	"How to Disable the Mounting of Lower-Level Files" on page 222
Troubleshoot file system mounting problems.	Resolve problems with mounting a file system.	"How to Troubleshoot Mount Failures in Trusted Extensions" on page 246

▼ How to Back Up Files in Trusted Extensions

1 Assume the Operator role.

This role includes the Media Backup rights profile.

2 Use one of the following backup methods:

- `/usr/lib/fs/ufs/ufsdump` for major backups
- `/usr/sbin/tar cT` for small backups
- A script calling either of these commands

  For example, the Budtool backup application calls the `ufsdump` command. See the `ufsdump(1M)` man page. For details on the `T` option to the `tar` command, see the `tar(1)` man page.

▼ How to Restore Files in Trusted Extensions

1. **Assume the System Administrator role.**

   This role includes the Media Restore rights profile.

2. **Use one of the following methods:**

   - `/usr/lib/fs/ufs/ufsrestore` for major restores
   - `/usr/sbin/tar xT` for small restores
   - A script calling either of these commands

   For details on the `T` option to the `tar` command, see the `tar(1)` man page.

Caution – Only these commands preserve labels.

▼ How to Share Directories From a Labeled Zone

As in the Solaris OS, the Mounts and Shares tool in the Solaris Management Console is used to share and mount files from the global zone. The tool cannot be used to mount or share directories that originate in labeled zones. Create a `dfstab` file at the label of the zone, and then restart the zone to share the labeled directories.

Caution – Do not use proprietary names for shared file systems. The names of shared file systems are visible to every user.

Before You Begin You must be superuser, or in the System Administrator role in the global zone on the file server.

1. **Create a workspace at the label of the directory that is going to be shared.**

   For details, see "How to Add a Workspace at a Particular Label" in *Solaris Trusted Extensions User's Guide*.

2. **Create a `dfstab` file in at the label of that zone.**

   For each zone that will share a directory, repeat the following steps:

   a. **Create the `/etc/dfs` directory in the zone.**

   ```
   # mkdir -p /zone/zone-name/etc/dfs
   ```

   b. **Open the trusted editor.**

   For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

c. **Type the full pathname of the `dfstab` file into the editor.**

```
# /zone/zone-name/etc/dfs/dfstab
```

d. **Add an entry to share a directory from that zone.**

The entry describes the directory from the perspective of the zone root path. For example, the following entry shares an application's files at the label of the containing zone:

```
share -F nfs -o ro /viewdir/viewfiles
```

3 **For each zone, share the directories by starting the zone.**

In the global zone, run one of the following commands for each zone. Each zone can share its directories in any of these ways. The actual sharing occurs when each zone is brought into the `ready` or `running` state.

- **If the zone is not in the running state and you do not want users to log in to the server at the label of the zone, set the zone state to `ready`.**

  ```
  # zoneadm -z zone-name ready
  ```

- **If the zone is not in the running state and users are allowed to log in to the server at the label of the zone, boot the zone.**

  ```
  # zoneadm -z zone-name boot
  ```

- **If the zone is already running, reboot the zone.**

  ```
  # zoneadm -z zone-name reboot
  ```

4 **Display the directories that are shared from your system.**

```
# showmount -e
```

5 **To enable the client to mount the exported files, see "How to NFS Mount Files in a Labeled Zone" on page 241.**

Example 17–2 Sharing the `/export/share` Directory at the `PUBLIC` Label

For applications that run at the label `PUBLIC`, the system administrator enables users to read the documentation in the `/export/share` directory of the `public` zone. The zone named `public` runs at the label `PUBLIC`.

First, the administrator creates a `public` workspace and edits the `dfstab` file.

```
# mkdir -p /zone/public/etc/dfs
# /usr/dt/bin/trusted_edit /zone/public/etc/dfs/dfstab
```

In the file, the administrator adds the following entry:

```
## Sharing PUBLIC user manuals
share -F nfs -o ro /export/appdocs
```

The administrator leaves the `public` workspace and returns to the Trusted Path workspace. Because users are not allowed to log in to this system, the administrator shares the files by putting the zone in the ready state:

```
# zoneadm -z public ready
```

Users can access the shared directories once the directories are mounted on the users' systems.

▼ How to NFS Mount Files in a Labeled Zone

In Trusted Extensions, a labeled zone manages the mounting of files in its zone.

Files from unlabeled and labeled hosts can be mounted on a Trusted Extensions labeled host.

- To mount the files read/write from a single-label host, the assigned label of the remote host must be identical to the zone in which the file is being mounted.
- Files that are mounted by a higher-level zone are read-only.
- In Trusted Extensions, the `auto_home` configuration file is customized per zone. The file is named by zone name. For example, a system with a global zone and a public zone has two `auto_home` files, `auto_home_global` and `auto_home_public`.

Trusted Extensions uses the same mounting interfaces as the Solaris OS:

- To mount files at boot, use the `/etc/vfstab` file in the labeled zone.
- To mount files dynamically, use the `mount` command in the labeled zone.
- To automount home directories, use the `auto_home_`*zone-name* files.
- To automount other directories, use the standard automount maps. If the automount maps are in LDAP, use LDAP commands to manage them.

Before You Begin You must be on the client system, in the zone at the label of the files that you want to mount. Unless you are using the automounter, you must be superuser, or be in the System Administrator role. To mount from lower-level servers, the zone must be configured with the `net_mac_aware` privilege.

- **To NFS mount files in a labeled zone, use the following procedures.**

  Most procedures include creating a workspace at a particular label. To create a workspace, see "How to Add a Workspace at a Particular Label" in *Solaris Trusted Extensions User's Guide*.

  - **Mount files dynamically.**

    In the labeled zone, use the `mount` command. For an example of mounting files dynamically, see Example 17–3.

- **Mount files when the zone boots**

  In the labeled zone, add the mounts to the `vfstab` file.

  For examples of mounting files when a labeled zone boots, see Example 17–4 and Example 17–5.

- **Mount home directories for systems that are administered with LDAP.**

  a. **At every label, add the user specifications to the** `auto_home_`*zone-name* **files.**

  b. **Then, use these files to populate the** `auto_home_`*zone-name* **database on the LDAP server.**

  For an example, see Example 17–6.

- **Mount home directories for systems that are administered with files.**

  a. **Create and populate an** `/export/home/auto_home_`*lowest-labeled-zone-name* **file.**

  b. **Edit the** `/etc/auto_home_`*lowest-labeled-zone-name* **file to point to the newly populated file.**

  c. **Modify the** `/etc/auto_home_`*lowest-labeled-zone-name* **file in every higher-level zone to point to the file that you created in Step a.**

  For an example, see Example 17–7.

Example 17–3 Mounting Files in a Labeled Zone by Using the `mount` Command

In this example, the system administrator mounts a remote file system from a public zone. The public zone is on a multilevel server.

After assuming the System Administrator role, the administrator creates a workspace at the label `PUBLIC`. In that workspace, the administrator runs the `mount` command.

```
# zonename
public
# mount -F nfs remote-sys:/zone/public/root/opt/docs  /opt/docs
```

A single-label file server at the label `PUBLIC` also contains documents to be mounted:

```
# mount -F nfs public-sys:/publicdocs  /opt/publicdocs
```

When the public zone of the `remote-sys` file server is in the `ready` or `running` state, the `remote-sys` files successfully mount on this system. When the `public-sys` file server is running, the files successfully mount.

Example 17–4 Mounting Files Read/Write in a Labeled Zone by Modifying the `vfstab` File

In this example, the system administrator mounts two remote file systems at the label `PUBLIC` in the local system's public zone when the public zone boots. One file system mount is from a multilevel system, and one file system mount is from a single-label system.

After assuming the System Administrator role, the administrator creates a workspace at the label `PUBLIC`. In that workspace, the administrator modifies the `vfstab` file in that zone.

```
## Writable books directories at PUBLIC
remote-sys:/zone/public/root/opt/docs  - /opt/docs  nfs  no  yes  rw
public-sys:/publicdocs    - /opt/publicdocs  nfs no yes rw
```

To access the files in the remote labeled zone of the multilevel system, the `vfstab` entry uses the zone root path of the remote system's public zone, `/zone/public/root`, as the directory pathname to the directories to mount. The path to the single-label system is identical to the path that would be used on a Solaris system.

In a terminal window at the label `PUBLIC`, the administrator mounts the files.

```
# mountall
```

Example 17–5 Mounting Lower-Level Files in a Labeled Zone by Modifying the `vfstab` File

In this example, the system administrator mounts a remote file system from a public zone in the local system's internal zone. After assuming the System Administrator role, the administrator creates a workspace at the label `INTERNAL`, then modifies the `vfstab` file in that zone.

```
## Readable books directory at PUBLIC
## ro entry indicates that PUBLIC docs can never be mounted rw in internal zone
remote-sys:/zone/public/root/opt/docs  - /opt/docs  nfs  no  yes  ro
```

To access the files in the remote labeled zone, the `vfstab` entry uses the zone root path of the remote system's public zone, `/zone/public/root`, as the directory pathname to the directories to mount.

From the perspective of a user in the internal zone, the files can be accessed at `/opt/docs`.

In a terminal window at the label `INTERNAL`, the administrator mounts the files.

```
# mountall
```

Example 17–6 Mounting Labeled Home Directories in a Network That Is Administered by Using LDAP

In this example, the system administrator enables a new user, `ikuk`, to access her home directory at every label. This site uses two home directory servers, and is administered by using LDAP. The second server contains the home directories for the users `jdoe` and `pkai`. The new user is added to this list.

First, after assuming the System Administrator role, the administrator modifies the `auto_home_`*zone-name* files in the `/etc` directory of the global zone to include the new user on the second home directory server.

```
## auto_home_global file
jdoe   homedir2-server:/export/home/jdoe
pkai   homedir2-server:/export/home/pkai
ikuk   homedir2-server:/export/home/ikuk
*      homedir-server:/export/home/&

## auto_home_internal file
## Mount the home directory from the internal zone of the NFS server
jdoe   homedir2-server:/export/home/jdoe
pkai   homedir2-server:/export/home/pkai
ikuk   homedir2-server:/export/home/ikuk
*      homedir-server:/export/home/&

## auto_home_public
## Mount the home directory from the public zone of the NFS server
jdoe   homedir2-server:/export/home/jdoe
pkai   homedir2-server:/export/home/pkai
ikuk   homedir2-server:/export/home/ikuk
*      homedir-server:/export/home/&
```

Next, to enable the users to log in at all labels, the administrator repeats these edits for the `auto_home_`*zone-name* files at every label.

Finally, after modifying every `auto_home_`*zone-name* file on this system, the administrator uses these files to add entries to the LDAP database.

Similar to the Solaris OS, the `+auto_home_public` entry in the `/etc/auto_home_`*zone-name* files directs the automounter to the LDAP entries. The `auto_home_`*zone-name* files on other systems on the network are updated from the LDAP database.

Example 17–7 Mounting a Lower-Level Home Directory on a System That Is Administered by Using Files

In this example, the system administrator enables users to access their home directories at every label. The labels at the site are `PUBLIC`, `INTERNAL`, and `NEEDTOKNOW`. This site uses two home directory servers, and is administered by using files. The second server contains the home directories for the users `jdoe` and `pkai`.

To accomplish this task, the system administrator defines the public zone NFS home directories in the public zone, and shares this configuration with the internal and needtoknow zones.

First, after assuming the System Administrator role, the administrator creates a workspace at the label `PUBLIC`. In this workspace, the administrator creates a new file, `/export/home/auto_home_public`. This file contains all the customized per-user NFS specification entries.

```
## /export/home/auto_home_public file at PUBLIC label
jdoe   homedir2-server:/export/home/jdoe
pkai   homedir2-server:/export/home/pkai
*      homedir-server:/export/home/&
```

Second, the administrator modifies the `/etc/auto_home_public` file to point to this new file.

```
## /etc/auto_home_public file in the public zone
## Use /export/home/auto_home_public for the user entries
## +auto_home_public
+ /export/home/auto_home_public
```

This entry directs the automounter to use the contents of the local file.

Third, the administrator similarly modifies the `/etc/auto_home_public` file in the internal and needtoknow zones. The administrator uses the pathname to the public zone that is visible to the internal and needtoknow zones.

```
## /etc/auto_home_public file in the internal zone
## Use /zone/public/export/home/auto_home_public for PUBLIC user home dirs
## +auto_home_public
+ /zone/public/export/home/auto_home_public

## /etc/auto_home_public file in the needtoknow zone
## Use /zone/public/export/home/auto_home_public for PUBLIC user home dirs
## +auto_home_public
+ /zone/public/export/home/auto_home_public
```

When the administrator adds the new user `ikuk`, the addition is made to the `/export/home/auto_home_public` file at the `PUBLIC` label.

```
## /export/home/auto_home_public file at PUBLIC label
jdoe   homedir2-server:/export/home/jdoe
pkai   homedir2-server:/export/home/pkai
ikuk   homedir2-server:/export/home/ikuk
*      homedir-server:/export/home/&
```

The higher-level zones read down to obtain the per-user home directories from the lower-level public zone.

▼ How to Troubleshoot Mount Failures in Trusted Extensions

Before You Begin You must be in the zone at the label of the files that you want to mount. You must be the superuser, or in the System Administrator role.

1. **Check the security attributes of the NFS server.**

   Use the Security Templates tool in the Solaris Management Console at the appropriate scope. For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

   a. **Verify that the IP address of the NFS server is an assigned host in one of the security templates.**

   The address might be directly assigned, or indirectly assigned through a wildcard mechanism. The address can be in a labeled template, or in an unlabeled template.

   b. **Check the label that the template assigns to the NFS server.**

   The label must be consistent with the label at which you are trying to mount the files.

2. **Check the label of the current zone.**

   If the label is higher than the label of the mounted file system, then you cannot write to the mount even if the remote file system is exported with read/write permissions. You can only write to the mounted file system at the label of the mount.

3. **To mount file systems from an NFS server that is running earlier versions of Trusted Solaris software, do the following:**

   - **For a Trusted Solaris 1 NFS server, use the `vers=2` and `proto=udp` options to the `mount` command.**
   - **For a Trusted Solaris 2.5.1 NFS server, use the `vers=2` and `proto=udp` options to the `mount` command.**

- **For a Trusted Solaris 8 NFS server, use the** `vers=3` **and** `proto=udp` **options to the** `mount` **command.**

To mount file systems from any of these servers, the server must be assigned to an unlabeled template.

CHAPTER 18

Trusted Networking (Overview)

This chapter describes trusted networking concepts and mechanisms in Trusted Extensions.

The Trusted Network

Trusted Extensions assigns security attributes to zones, hosts, and networks. These attributes ensure that the following security features are enforced on the network:

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

In Trusted Extensions, network packets are protected by MAC. Labels are used for MAC decisions. Data is labeled explicitly or implicitly with a sensitivity label. A label has an ID field, a classification or "level" field, and a compartment or "category" field. Data must pass an accreditation check. This check determines if the label is well formed, and if the label lies within the accreditation range of the receiving host. Well-formed packets that are within the receiving host's accreditation range are granted access.

IP packets that are exchanged between trusted systems can be labeled. Trusted Extensions supports Commercial IP Security Option (CIPSO) labels. A CIPSO label on a packet serves to classify, segregate, and route IP packets. Routing decisions compare the sensitivity label of the data with the label of the destination.

Typically on a trusted network, the label is generated by a sending host and processed by the receiving host. However, a trusted router can also add or strip labels while forwarding packets in a trusted network. A sensitivity label is mapped to a CIPSO label before transmission. The CIPSO label is embedded in the IP packet. Typically, a packet sender and the packet's receiver operate at the same label.

Trusted networking software ensures that the Trusted Extensions security policy is enforced even when the subjects (processes) and objects (data) are located on different hosts. Trusted Extensions networking preserves MAC across distributed applications.

Trusted Extensions Data Packets

Trusted Extensions data packets include a CIPSO label option. The data packets can be sent over IPv4 or IPv6 networks.

In the standard IPv4 format, the IPv4 header with options is followed by a TCP, UDP, or SCTP header and then the actual data. The Trusted Extensions version of an IPv4 packet uses the CIPSO option in the IP header for the security attributes.

IPv4 Header With CIPSO Option	TCP, UDP, or SCTP	Data

In the standard IPv6 format, an IPv6 header with extensions is followed by a TCP, UDP, or SCTP header and then the actual data. The Trusted Extensions IPv6 packet includes a multilevel security option in the header with extensions.

IPv6 Header With Extensions	TCP, UDP, or SCTP	Data

Trusted Network Communications

Trusted Extensions supports labeled and unlabeled hosts on a trusted network. LDAP is a fully supported naming service. Various commands and GUIs enable the network to be administered.

Systems that run Trusted Extensions software support network communications between Trusted Extensions hosts and any of the following types of systems:

- Other systems that are running Trusted Extensions
- Systems that are running operating systems that do not recognize security attributes, but do support TCP/IP, such as Solaris systems, other UNIX® systems, Microsoft Windows, and Macintosh OS systems

- Systems that are running other trusted operating systems that recognize CIPSO labels

As in the Solaris OS, Trusted Extensions network communications and services can be managed by a naming service. Trusted Extensions adds the following interfaces to Solaris network interfaces:

- Trusted Extensions adds three network configuration databases, `tnzonecfg`, `tnrhdb`, and `tnrhtp`. For details, see "Network Configuration Databases in Trusted Extensions" on page 252.
- The Trusted Extensions version of the naming service switch file, `nsswitch.conf`, includes entries for the `tnrhtp` and `tnrhdb` databases. These entries can be modified to suit each site's configuration.

  Trusted Extensions uses the LDAP naming service to centrally manage configuration files that define hosts, networks, and users. The default `nsswitch.conf` entries for the trusted network databases for the LDAP naming service follow:

  ```
  # Trusted Extensions
  tnrhtp: files ldap
  tnrhdb: files ldap
  ```

  The LDAP naming service on a Sun Java System Directory Server is the only fully supported naming service in Trusted Extensions. For information about the use of LDAP on a system that is configured with Trusted Extensions, see Chapter 15, "Trusted Extensions and LDAP (Overview)."
- Trusted Extensions adds tools to the Solaris Management Console. The console is used to centrally manage zones, hosts, and networks. The network tools are described in "Solaris Management Console Tools" on page 141.

  The Part I, "Initial Configuration of Trusted Extensions," describes how to define zones and hosts when you configure the network. For additional details, see Chapter 19, "Managing Networks in Trusted Extensions (Tasks)."
- Trusted Extensions adds commands to administer trusted networking. Trusted Extensions also adds options to the Solaris network commands. For a description of these commands, see "Network Commands in Trusted Extensions" on page 252.
- Trusted Extensions extends the IKE configuration file, `/etc/inet/ike/config` for labeled IPsec. For more information, see "Administration of Labeled IPsec" on page 264 and the `ike.config(4)` man page

Network Configuration Databases in Trusted Extensions

Trusted Extensions loads three network configuration databases into the kernel. These databases are used in accreditation checks as data is transmitted from one host to another host.

- `tnzonecfg` – This local database stores zone attributes that are security-related. The attributes for each zone specify the zone label and the zone's access to single-level and multilevel ports. Another attribute handles responses to control messages, such as `ping`. The labels for zones are defined in the `label_encodings` file. For more information, see the `label_encodings(4)` and `smtnzonecfg(1M)` man pages. For a discussion of multilevel ports, see "Zones and Multilevel Ports" on page 215.
- `tnrhtp` – This database stores templates that describe the security attributes of hosts and gateways. `tnrhtp` can be a local database or stored on the LDAP server. Hosts and gateways use the attributes of the destination host and next-hop gateway to enforce MAC when sending traffic. When receiving traffic, hosts and gateways use the attributes of the sender. For details of the security attributes, see "Trusted Network Security Attributes" on page 253. For more information, see the `smtnrhtp(1M)` man page.
- `tnrhdb` – This database holds the IP addresses and network prefixes (fallback mechanism) that correspond to all hosts that are allowed to communicate. `tnrhdb` can be a local database or stored on the LDAP server. Each host or network prefix is assigned a security template from the `tnrhtp` database. The attributes in the template define the attributes of the assigned host. For more information, see the `smtnrhdb(1M)` man page.

In Trusted Extensions, the Solaris Management Console has been extended to handle these databases. For details, see "Solaris Management Console Tools" on page 141.

Network Commands in Trusted Extensions

Trusted Extensions adds the following commands to administer trusted networking:

- `tnchkdb` – This command is used to verify the correctness of the trusted network databases. The `tnchkdb` command is used whenever you change a security template (`tnrhtp`), a security template assignment (`tnrhdb`), or the configuration of a zone (`tnzonecfg`). The Solaris Management Console tools run this command automatically when a database is modified. For details, see the `tnchkdb(1M)` man page.
- `tnctl` – This command can be used to update the trusted network information in the kernel. `tnctl` is also a system service. A restart with the command `svcadm restart /network/tnctl` refreshes the kernel cache from the trusted network databases on the local system. The Solaris Management Console tools run this command automatically when a database is modified in the Files scope. For details, see the `tnctl(1M)` man page.

- `tnd` – This daemon pulls `tnrhdb` and `tnrhtp` information from the LDAP directory. `tnd` is started at boot time as a service, as in `svcadm start /network/tnd`. This command also can be used for debugging and for changing the polling interval. For details, see the `tnd(1M)` man page.
- `tninfo` – This command displays the details of the current state of the trusted network kernel cache. The output can be filtered by host name, zone, or security template. For details, see the `tninfo(1M)` man page.

Trusted Extensions adds options to the following Solaris network commands:

- `ifconfig` – The `all-zones` interface flag for this command makes the specified interface available to every zone on the system. The appropriate zone to deliver data to is determined by the label that is associated with the data. For details, see the `ifconfig(1M)` man page.
- `netstat` – The `-R` option extends Solaris `netstat` usage to display Trusted Extensions-specific information, such as security attributes for multilevel sockets and routing table entries. The extended security attributes include the label of the peer, and whether the socket is specific to a zone, or available to several zones. For details, see the `netstat(1M)` man page.
- `route` – The `-secattr` option extends Solaris `route` usage to display the security attributes of the route. The value of the option has the following format:

  `min_sl=`*label*`,max_sl=`*label*`,doi=`*integer*`,cipso`

  The `cipso` keyword is optional and set by default. For details, see the `route(1M)` man page.
- `snoop` – As in the Solaris OS, the `-v` option to this command can be used to display the IP headers in detail. In Trusted Extensions, the headers contain label information.
- `ipseckey` – In Trusted Extensions, the following extensions are available to label IPsec-protected packets: `label` *label*, `outer-label` *label*, and `implicit-label` *label*. For details, see the `ipseckey(1M)` man page.

Trusted Network Security Attributes

Network administration in Trusted Extensions is based on security templates. A security template describes a set of hosts that have common protocols and identical security attributes.

Security attributes are administratively assigned to systems, both hosts and routers, by means of templates. The security administrator administers templates and assigns them to systems. If a system does not have an assigned template, no communications are allowed with that system.

Every template is named, and includes the following:

- A host type of either Unlabeled or CIPSO. The protocol that is used for network communications is determined by the host type of the template.

  The host type is used to determine whether to use CIPSO options and affects MAC. See "Host Type and Template Name in Security Templates" on page 255.

- A set of security attributes that are applied to each host type.

For more detail about host types and security attributes, see "Network Security Attributes in Trusted Extensions" on page 254.

Network Security Attributes in Trusted Extensions

Trusted Extensions is installed with a default set of security templates. When a template is assigned to a host, the security values in the template are applied to the host. In Trusted Extensions, both unlabeled hosts and labeled hosts on the network are assigned security attributes by means of a template. Hosts that are not assigned a security template cannot be reached. The templates can be stored locally, or in the LDAP naming service on the Sun Java System Directory Server.

Templates can be assigned directly or indirectly to a host. Direct assignment assigns a template to a particular IP address. Indirect assignment assigns a template to a network address that includes the host. Hosts that do not have a security template cannot communicate with hosts that are configured with Trusted Extensions. For an explanation of direct assignment and indirect assignment, see "Trusted Network Fallback Mechanism" on page 257.

Templates are modified or created by using the Security Templates tool in the Solaris Management Console. The Security Templates tool enforces the completion of the required fields in the templates. Which fields are required is based on the host type.

Each host type has its own set of additional required and optional security attributes. The following security attributes are specified in security templates:

- **Host type** – Defines whether the packets are labeled with CIPSO security labels or not labeled at all.
- **Default label** – Defines the level of trust of the unlabeled host. Packets that are sent by an unlabeled host are read at this label by the receiving Trusted Extensions host or gateway.

  The Default label attribute is specific to the unlabeled host type. For details, see the `smtnrhtp`(1M) man page and the following sections.
- **DOI** – A positive, non-zero integer that identifies the domain of interpretation. The DOI is used to indicate which set of label encodings applies to a network communication or network entity. Labels with different DOIs, even if otherwise identical, are disjoint. For unlabeled hosts, the DOI applies to the default label. In Trusted Extensions, the default value is `1`.
- **Minimum label** – Defines the bottom of the label accreditation range. Hosts and next-hop gateways do not receive packets that are below the minimum label that is specified in their template.
- **Maximum label** – Defines the top of the label accreditation range. Hosts and next-hop gateways do not receive packets that are higher than the maximum label that is specified in their template.

- **Security label set** – Optional. Specifies a discrete set of security labels for a security template. In addition to their accreditation range that is determined by the maximum and minimum label, hosts that are assigned to a template with a security label set can send and receive packets that match any one of the labels in the label set. The maximum number of labels that can be specified is four.

Host Type and Template Name in Security Templates

Trusted Extensions supports two host types in the trusted network databases and provides two default templates:

- **CIPSO host type –** Intended for hosts that run trusted operating systems. Trusted Extensions supplies the template named `cipso` for this host type.

  The Common IP Security Option (CIPSO) protocol is used to specify security labels that are passed in the IP options field. CIPSO labels are derived automatically from the data's label. Tag type 1 is used to pass the CIPSO security label. This label is then used to make security checks at the IP level and to label the data in the network packet.
- **Unlabeled host type** - Intended for hosts that use standard networking protocols but do not support CIPSO options. Trusted Extensions supplies the template named `admin_low` for this host type.

  This host type is assigned to hosts that run the Solaris OS or other unlabeled operating systems. This host type gives provides a default label and a default clearance to apply to communications with the unlabeled host. Also, a label range or a set of discrete labels can be specified to allow the sending of packets to an unlabeled gateway for forwarding.

Caution – The `admin_low` template provides an example for constructing unlabeled templates with site-specific labels. While the `admin_low` template is required for the installation of Trusted Extensions, the security settings might not be appropriate for normal system operations. Retain the provided templates without modification for system maintenance and support reasons.

Default Label in Security Templates

Templates for the unlabeled host type specify a default label. This label is used to control communications with hosts whose operating systems are not aware of labels, such as Solaris systems. The default label that is assigned reflects the level of trust that is appropriate for the host and its users.

Because communications with unlabeled hosts are essentially limited to the default label, these hosts are also referred to as *single-label hosts*.

Domain of Interpretation in Security Templates

Organizations that use the same Domain of Interpretation (DOI) agree among themselves to interpret label information and other security attributes in the same way. When Trusted Extensions performs a label comparison, a check is made as to whether the DOI is equal.

A Trusted Extensions system enforces label policy on one DOI value. All zones on a Trusted Extensions system must operate at the same DOI. A Trusted Extensions system does not provide exception handling on packets that are received from a system that uses a different DOI.

If your site uses a DOI value that is different from the default value, you must add this value to the `/etc/system` file, and change the value in every security template. For the initial procedure, see "Configure the Domain of Interpretation" on page 57. To configure the DOI in every security template, see Example 19–1.

Label Range in Security Templates

The minimum label and maximum label attributes are used to establish the label range for labeled and unlabeled hosts. These attributes are used to do the following:

- To set the range of labels that can be used when communicating with a remote CIPSO host

  In order for a packet to be sent to a destination host, the label of the packet must be within the label range assigned to the destination host in the security template for that host.
- To set a label range for packets that are being forwarded through a CIPSO gateway or an unlabeled gateway

  The label range can be specified in the template for an unlabeled host type. The label range enables the host to forward packets that are not necessarily at the label of the host, but are within a specified label range.

Security Label Set in Security Templates

The security label set defines at most four discrete labels at which packets can be accepted, forwarded, or sent by the remote host. This attribute is optional. By default, no security label set is defined.

Trusted Network Fallback Mechanism

The tnrhdb database can assign a security template to a particular host either directly or indirectly. Direct assignment assigns a template to a host's IP address. Indirect assignment is handled by a fallback mechanism. The trusted network software first looks for an entry that specifically assigns the host's IP address to a template. If the software does not find a specific entry for the host, it looks for the “longest prefix of matching bits”. You can indirectly assign a host to a security template when the IP address of the host falls within the “longest prefix of matching bits” of an IP address with a fixed prefix length.

In IPv4, you can make an indirect assignment by subnet. When you make an indirect assignment by using 4, 3, 2, or 1 trailing zero (0) octets, the software calculates a prefix length of 0, 8, 16, or 24, respectively. Entries 3 – 6 in Table 18–1 illustrate this fallback mechanism.

You can also set a fixed prefix length by adding a slash (/) followed by the number of fixed bits. IPv4 network addresses can have a prefix length between 1 – 32. IPv6 network addresses can have a prefix length between 1 – 128.

The following table provides fallback address and host address examples. If an address within the set of fallback addresses is directly assigned, the fallback mechanism is not used for that address.

TABLE 18–1 `tnrhdb` Host Address and Fallback Mechanism Entries

IP Version	`tnrhdb` Entry	Addresses Covered
IPv4	`192.168.118.57:cipso`	`192.168.118.57`
	`192.168.118.57/32:cipso`	The `/32` sets a prefix length of 32 fixed bits.
	`192.168.118.128/26:cipso`	From `192.168.118.0` through `192.168.118.63`
	`192.168.118.0:cipso`	All addresses on `192.168.118.` network
	`192.168.118.0/24:cipso`	
	`192.168.0.0/24:cipso`	All addresses on `192.168.0.` network.
	`192.168.0.0:cipso`	All addresses on `192.168.` network
	`192.168.0.0/16:cipso`	
	`192.0.0.0:cipso`	All addresses on `192.` network
	`192.0.0.0/8:cipso`	
	`192.168.0.0/32:cipso`	Network address `192.168.0.0`. Not a wildcard address.
	`192.168.118.0/32:cipso`	Network address `192.168.118.0`. Not a wildcard address.
	`192.0.0.0/32:cipso`	Network address `192.0.0.0`. Not a wildcard address.
	`0.0.0.0/32:cipso`	Host address `0.0.0.0`. Not a wildcard address.
	`0.0.0.0:cipso`	All addresses on all networks
IPv6	`2001\:DB8\:22\:5000\:\:21f7:cipso`	`2001:DB8:22:5000::21f7`
	`2001\:DB8\:22\:5000\:\:0/52:cipso`	From `2001:DB8:22:5000::0` through `2001:DB8:22:5fff:ffff:ffff:ffff:ffff`
	`0\:\:0/0:cipso`	All addresses on all networks

Note that the `0.0.0.0/32` address matches the specific address, `0.0.0.0`. The `tnrhdb` entry `0.0.0.0/32:admin_low` is useful on a system where the literal address, `0.0.0.0`, is used as a source IP address. For example, DHCP clients contact the DHCP server as `0.0.0.0` before the server provides the clients with an IP address.

To create a `tnrhdb` entry on a Sun Ray server that serves DHCP clients, see Example 19–13. Because `0.0.0.0:admin_low` is the default wildcard entry, see "How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 279 for issues to consider before removing or changing this default.

For more information about prefix lengths in IPv4 and IPv6 addresses, see "Designing Your CIDR IPv4 Addressing Scheme" in *System Administration Guide: IP Services* and "IPv6 Addressing Overview" in *System Administration Guide: IP Services*.

Overview of Routing in Trusted Extensions

In Trusted Extensions, routes between hosts on different networks must maintain security at each step in the transmission. Trusted Extensions adds extended security attributes to the routing protocols in the Solaris OS. Unlike the Solaris OS, this Trusted Extensions release does not support dynamic routing. For details about specifying static routing, see the `-p` option in the `route`(1M) man page.

Gateways and routers route packets. In this discussion, the terms "gateway" and "router" are used interchangeably.

For communications between hosts on the same subnet, accreditation checks are performed at endpoints only because no routers are involved. Label range checks are performed at the source. If the receiving host is running Trusted Extensions software, label range checks are also performed at the destination.

When the source and destination hosts are on different subnets, the packet is sent from the source host to a gateway. The label range of the destination and the first-hop gateway is checked at the source when a route is selected. The gateway forwards the packet to the network where the destination host is connected. A packet might go through several gateways before reaching the destination.

Background on Routing

On Trusted Extensions gateways, label range checks are performed in certain cases. A Trusted Extensions system that is routing a packet between two unlabeled hosts compares the default label of the source host to the default label of the destination host. When the unlabeled hosts share a default label, the packet is routed.

Each gateway maintains a list of routes to all destinations. Standard Solaris routing makes choices to optimize the route. Trusted Extensions provides additional software to check security requirements that apply to the route choices. The Solaris choices that do not satisfy security requirements are skipped.

Routing Table Entries in Trusted Extensions

The routing table entries in Trusted Extensions can incorporate security attributes. Security attributes can include a `cipso` keyword. Security attributes must include a maximum label, a minimum label, and a DOI.

For entries that do not provide security attributes, the attributes in the gateway's security template are used.

Trusted Extensions Accreditation Checks

Trusted Extensions software determines the suitability of a route for security purposes. The software runs a series of tests called *accreditation checks* on the source host, the destination host, and the intermediate gateways.

Note – In the following discussion, an accreditation check for a label range also means a check for a security label set.

The accreditation check verifies the label range and CIPSO label information. The security attributes for a route are obtained from the routing table entry, or from the security template of the gateway if the entry has no security attributes.

For incoming communications, the Trusted Extensions software obtains labels from the packets themselves, whenever possible. Obtaining labels from packets is only possible when the messages are sent from systems that support labels. When a label is not available from the packet, a default label is assigned to the message from trusted networking database files. These labels are then used during accreditation checks. Trusted Extensions enforces several checks on outgoing messages, forwarded messages, and incoming messages.

Source Accreditation Checks

The following accreditation checks are performed on the sending process or sending zone:

- For all destinations, the label of the data must be within the label range of the next hop in the route, that is, the first hop. And, the label must be contained in the first-hop gateway's security attributes.
- For all destinations, the DOI of an outgoing packet must match the DOI of the destination host. The DOI must also match the DOI of all hops along the route, including its first-hop gateway.
- When the destination host is an unlabeled host, one of the following conditions must be satisfied:
  - The sending host's label must match the destination host's default label.
  - The sending host is privileged to perform cross-label communication, and the sender's label dominates the destination's default label.
  - The sending host is privileged to perform cross-label communication, and the sender's label is `ADMIN_LOW`. That is, the sender is sending from the global zone.

Note – A first-hop check occurs when a message is being sent through a gateway from a host on one network to a host on another network.

Gateway Accreditation Checks

On a Trusted Extensions gateway system,the following accreditation checks are performed for the next-hop gateway:

- If the incoming packet is unlabeled, the packet inherits the source host's default label from the `tnrhdb` entry. Otherwise, the packet receives the indicated CIPSO label.
- Checks for forwarding a packet proceed similar to source accreditation:
  - For all destinations, the label of the data must be within the label range of the next hop. And, the label must be contained in the security attributes of the next-hop host.
  - For all destinations, the DOI of an outgoing packet must match the DOI of the destination host. The DOI must also match the DOI of the next-hop host.
  - The label of an unlabeled packet must match the destination host's default label.
  - The label of a CIPSO packet must be within the destination host's label range.

Destination Accreditation Checks

When a Trusted Extensions host receives data, the software performs the following checks:

- If the incoming packet is unlabeled, the packet inherits the source host's default label from the `tnrhdb` entry. Otherwise, the packet receives the indicated CIPSO label.
- The label and DOI for the packet must be consistent with the destination zone or destination process's label and DOI. The exception is when a process is listening on a multilevel port. The listening process can receive a packet if the process is privileged to perform cross-label communications, and the process is either in the global zone or has a label that dominates the packet's label.

Administration of Routing in Trusted Extensions

Trusted Extensions supports several methods for routing communications between networks. In the Security Administrator role, you can set up routes that enforce the degree of security required by your site's security policy.

For example, sites can restrict communications outside the local network to a single label. This label is applied to publicly available information. Labels such as `UNCLASSIFIED` or `PUBLIC` can indicate public information. To enforce the restriction, these sites assign a single-label template to the network interface that is connected to the external network. For more details about TCP/IP and routing, see the following:

- "Planning for Routers on Your Network" in *System Administration Guide: IP Services*
- "Configuring Systems on the Local Network" in *System Administration Guide: IP Services*
- "Major TCP/IP Administrative Tasks (Task Map)" in *System Administration Guide: IP Services*
- "Preparing Your Network for the DHCP Service (Task Map)" in *System Administration Guide: IP Services*

Choosing Routers in Trusted Extensions

Trusted Extensions hosts offer the highest degree of trust as routers. Other types of routers might not recognize Trusted Extensions security attributes. Without administrative action, packets can be routed through routers that do not provide MAC security protection.

- CIPSO routers drop packets when they do not find the correct type of information in the IP options section of the packet. For example, a CIPSO router drops a packet if it does not find a CIPSO option in the IP options when the option is required, or when the DOI in the IP options is not consistent with the destination's accreditation.
- Other types of routers that are not running Trusted Extensions software can be configured to either pass the packets or drop the packets that include the CIPSO option. Only CIPSO-aware gateways such as Trusted Extensions provides can use the contents of the CIPSO IP option to enforce MAC.

To support trusted routing, the Solaris Express Community Edition routing tables are extended to include Trusted Extensions security attributes. The attributes are described in "Routing Table Entries in Trusted Extensions" on page 259. Trusted Extensions supports static routing, in which the administrator creates routing table entries manually. For details, see the `-p` option in the `route(1M)` man page.

The routing software tries to find a route to the destination host in the routing tables. When the host is not explicitly named, the routing software looks for an entry for the subnetwork where the host resides. When neither the host nor the network where the host resides is defined, the host sends the packet to a default gateway, if defined. Multiple default gateways can be defined, and each is treated equally.

In this release of Trusted Extensions, the security administrator sets up routes manually, and then manually changes the routing table when conditions change. For example, many sites have a single gateway that communicates with the outside world. In these cases, the single gateway can be statically defined as the *default* on each host on the network. Dynamic routing support might be available in future releases of Trusted Extensions.

Gateways in Trusted Extensions

An example of routing in Trusted Extensions follows. The diagram and table show three potential routes between Host 1 and Host 2.

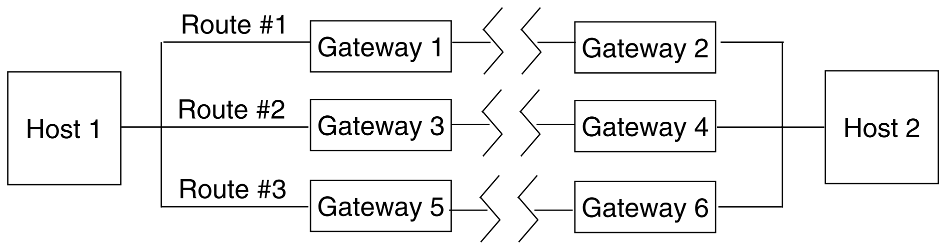


FIGURE 18–1 Typical Trusted Extensions Routes and Routing Table Entries

Route	First-Hop Gateway	Minimum Label	Maximum Label	DOI
#1	Gateway 1	`CONFIDENTIAL`	`SECRET`	1
#2	Gateway 3	`ADMIN_LOW`	`ADMIN_HIGH`	1
#3	Gateway 5			

- Route #1 can transmit packets within the label range of `CONFIDENTIAL` to `SECRET`.
- Route #2 can transmit packets from `ADMIN_LOW` to `ADMIN_HIGH`.
- Route #3 does not specify routing information. Therefore, its security attributes are derived from the template in the `tnrhtp` database for Gateway 5.

Routing Commands in Trusted Extensions

To show labels and extended security attributes for sockets, Trusted Extensions modifies the following Solaris network commands:

- The `netstat -rR` command displays the security attributes in routing table entries.
- The `netstat -aR` command displays the security attributes for sockets.
- The `route -p` command with the `add` or `delete` option changes the routing table entries.

For details, see the `netstat`(1M) and `route`(1M) man pages.

For examples, see "How to Configure Routes With Security Attributes" on page 284.

Administration of Labeled IPsec

Solaris Trusted Extensions systems can protect labeled network packets with IPsec. The IPsec packets can be sent with explicit or implicit Trusted Extensions labels. Labels are sent explicitly by using CIPSO IP options and implicitly by using labeled IPsec security associations (SAs). Also, IPsec encrypted packets with different implicit labels can be tunneled across an unlabeled network.

For general IPsec concepts and configuration procedures, see Part III, "IP Security," in *System Administration Guide: IP Services*. For Trusted Extensions modifications to IPsec procedures, see "Configuring Labeled IPsec (Task Map)" on page 289.

Labels for IPsec-Protected Exchanges

All communications on Trusted Extensions systems, including IPsec-protected communications, must satisfy security label accreditation checks. The checks are described in "Trusted Extensions Accreditation Checks" on page 260.

The labels on IPsec packets that must pass these checks are the *inner label*, the *wire label*, and the *key management label*:

- **Application security label –** The label of the zone in which the application resides.
- **Inner label –** The label of the unencrypted message data before IPsec AH or ESP headers have been applied. This label can be different from the application security label when the `SO_MAC_EXEMPT` socket option (MAC-exempt) or multilevel port (MLP) features are being used. When selecting security associations (SAs) and IKE rules that are constrained by labels, IPsec and IKE use this inner label.

  By default, the inner label is the same as the application security label. Typically, applications at both ends have the same label. However, for MAC-exempt or MLP communication, this condition might not be true. IPsec configuration settings can define how the inner label is conveyed across the network, that is, they can define the *wire label*. IPsec configuration settings cannot define the value of the inner label.
- **Wire label –** The label of the encrypted message data after IPsec AH or ESP headers have been applied. Depending on the IKE and IPsec configuration files, the wire label might be different from the inner label.
- **Key management label –** All IKE negotiations between two nodes are controlled at a single label, regardless of the label of application messages that trigger the negotiations. The label of IKE negotiations is defined in the `/etc/inet/ike/config` file on a per-IKE rule basis.

Label Extensions for IPsec Security Associations

IPsec *label extensions* are used on Trusted Extensions systems to associate a label with the traffic that is carried inside a security association (SA). By default, IPsec does not use label extensions and therefore ignores labels. All traffic between two systems flows through a single SA, regardless of the Trusted Extensions label.

Label extensions enable you to do the following:

- Configure a different IPsec SA for use with each Trusted Extensions label. This configuration effectively provides an additional mechanism for conveying the label of traffic that travels between two multilevel systems.
- Specify an on-the-wire label for IPsec encrypted message text that is different from the unencrypted form of the text. This configuration supports the transmission of encrypted confidential data through a less secure network.
- Suppress the use of CIPSO IP options in IP packets. This configuration enables labeled traffic to traverse CIPSO-unaware or CIPSO-hostile networks.

You can specify whether to use label extensions automatically through IKE as described in "Label Extensions for IKE" on page 265, or manually through the `ipseckey` command. For details on the label extensions features, see the `ipseckey(1M)` man page.

When using label extensions, SA selection for outbound traffic includes the inner sensitivity label as part of the match. The security label of inbound traffic is defined by the security label of received packet's SA.

Label Extensions for IKE

IKE on Trusted Extensions systems supports the negotiation of labels for SAs with label-aware peers.

You can control this mechanism by using the following keywords in the `/etc/inet/ike/config` file:

- **label_aware** – Enables the `in.iked` daemon's use of Trusted Extensions label interfaces and the negotiation of labels with peers.
- **single_label** – Indicates that the peer does not support the negotiation of labels for SAs.
- **multi_label** – Indicates that the peer supports the negotiation of labels for SAs. IKE creates a new SA for each additional label that IKE encounters in the traffic between two nodes.
- **wire_label inner** – Causes the `in.iked` daemon to create labeled SAs where the wire label is the same as the inner label. The key management label is `ADMIN_LOW` when the daemon is negotiating with `cipso` peers. The key management label is the peer's default label when the daemon is negotiating with unlabeled peers. Normal Trusted Extensions rules are followed for inclusion of the CIPSO IP options in transmitted packets.

- **wire_label** *label* – Causes the `in.iked` daemon to create labeled SAs where the wire label is set to *label*, regardless of the value of the inner label. The `in.iked` daemon performs key management negotiations at the specified label. Normal Trusted Extensions rules are followed for inclusion of CIPSO IP options in transmitted packets.
- **wire_label none** *label* – Causes behavior similar to `wire_label` *label*, except that CIPSO IP options are suppressed on transmitted IKE packets and data packets under the SA.

For more information, see the `ike.config(4)` man page.

Labels and Accreditation in Tunnel Mode IPsec

When application data packets are protected by IPsec in tunnel mode, the packets contain multiple IP headers.

Outer IP Header	ESP or AH	Inner IP Header	TCP Header	Data

The IKE protocol's IP header contains the same source and destination address pair as the application data packet's outer IP header.

Outer IP Header	UDP Header	IKE Key Management Protocol

Trusted Extensions uses the inner IP header addresses for inner label accreditation checks. Trusted Extensions performs wire and key management label checks by using the outer IP header addresses. For information about the accreditation checks, see "Trusted Extensions Accreditation Checks" on page 260.

Confidentiality and Integrity Protections With Label Extensions

The following table explains how IPsec confidentiality and integrity protections apply to the security label with various configurations of label extensions.

Security Association	Confidentiality	Integrity
Without label extensions	Label is visible in the CIPSO IP option.	Message label in the CIPSO IP option is covered by AH, not by ESP. See Note.

Security Association	Confidentiality	Integrity
With label extensions	A CIPSO IP option is visible, but represents the wire label, which might be different from the inner message label.	Label integrity is implicitly covered by the existence of a label-specific SA. On-the-wire CIPSO IP option is covered by AH. See Note.
With label extensions and CIPSO IP option suppressed	Message label is not visible.	Label integrity is implicitly covered by existence of a label-specific SA.

Note – You cannot use IPsec AH integrity protections to protect the CIPSO IP option if CIPSO-aware routers might strip or add the CIPSO IP option as a message travels through the network. Any modification to the CIPSO IP option will invalidate the message and cause a packet that is protected by AH to be dropped at the destination.

CHAPTER 19

Managing Networks in Trusted Extensions (Tasks)

This chapter provides implementation details and procedures for securing a Solaris Trusted Extensions network.

Managing the Trusted Network (Task Map)

The following table points to the task maps for common trusted networking procedures.

Task	Description	For Instructions
Configure network databases.	Creates remote host templates,and assigns hosts to the templates.	"Configuring Trusted Network Databases (Task Map)" on page 270
Configure routing, and check network databases and network information in the kernel.	Configures static routes that enable labeled packets to reach their destination through labeled and unlabeled gateways. Also, displays the state of your network.	"Configuring Routes and Checking Network Information in Trusted Extensions (Task Map)" on page 283
Troubleshoot networking problems.	Steps to take when diagnosing network problems with labeled packets.	"Troubleshooting the Trusted Network (Task Map)" on page 294

Configuring Trusted Network Databases (Task Map)

Trusted Extensions software includes the `tnrhtp` and `tnrhdb` databases. These databases provide labels for remote hosts that contact the system. The Solaris Management Console provides the GUI that you use to administer these databases.

Task	Description	For Instructions
Determine if your site requires customized security templates.	Evaluates the existing templates for the security requirements of your site.	"How to Determine If You Need Site-Specific Security Templates" on page 271
Access the Security Templates tool in the Solaris Management Console.	Accesses the tool for modifying trusted network databases.	"How to Open the Trusted Networking Tools" on page 272
Modify security templates.	Modifies the definitions of security attributes in your trusted network by modifying the trusted network databases.	"How to Construct a Remote Host Template" on page 272
	Changes the DOI to a value different from `1`.	Example 19–1
	Creates a security template for labeled hosts that restrict communication between other hosts to a single label.	Example 19–2
	Creates a security template for unlabeled hosts that operate as single-label gateways.	Example 19–3
	Creates a security template for hosts with a restricted label range.	Example 19–4
	Creates a security template for a host that specifies a set of discrete labels in its label range.	Example 19–5
	Creates a security template for unlabeled systems and networks.	Example 19–6
	Creates a security template for two developer systems.	Example 19–7
Add hosts to the known network.	Adds systems and networks to the trusted network.	"How to Add Hosts to the System's Known Network" on page 277
Provide remote host access by using wildcard entries.	Allows hosts within a range of IP addresses to communicate with a system by indirectly assigning each host to the same security template.	Example 19–8 Example 19–9 Example 19–10

Task	Description	For Instructions
Change the admin_low wildcard entry in the tnrhdb file.	Increases security by replacing the wildcard entry with specific addresses for the host to contact at boot time.	"How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 279
	Increases security by replacing the wildcard entry with a network of labeled hosts as the default.	Example 19–11
Create an entry for the host address 0.0.0.0	Configures a Sun Ray server to accept the initial contact from a remote client	Example 19–13
Assign security templates.	Associates a template with an IP address or list of contiguous IP addresses.	"How to Assign a Security Template to a Host or a Group of Hosts" on page 278

▼ How to Determine If You Need Site-Specific Security Templates

Before You Begin You must be in the Security Administrator role in the global zone.

1 Familiarize yourself with the Trusted Extensions templates.

Read the `tnrhtp` file on a local host. The comments in the file are helpful. You can also view the security attribute values in the Security Templates tool in the Solaris Management Console.

- The default templates match any installation. The label range for each template is `ADMIN_LOW` to `ADMIN_HIGH`.
- The `cipso` template defines a CIPSO host type whose DOI is `1`. The label range for the template is `ADMIN_LOW` to `ADMIN_HIGH`.
- The `admin_low` template defines an unlabeled host whose DOI is `1`. The template's default label is `ADMIN_LOW`. The label range for the template is `ADMIN_LOW` to `ADMIN_HIGH`. In the default configuration, the address `0.0.0.0` is assigned to this template. Therefore, all non-CIPSO hosts are treated as hosts that operate at the `ADMIN_LOW` security label.

2 Keep the default templates.

For support purposes, do not delete or modify the default templates. You can change the host that is assigned these default templates. For an example, see "How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 279.

3 Create new templates if you want to do any of the following:

- Limit the label range of a host or a group of hosts.
- Create a single-label host.
- Create a host that recognizes a few discrete labels.
- Use a different DOI than `1`.

- Require a default label for unlabeled hosts that is not `ADMIN_LOW`.

For details, see "How to Construct a Remote Host Template" on page 272.

▼ How to Open the Trusted Networking Tools

Before You Begin You must be in the global zone in a role that can modify network security. For example, roles that are assigned the Information Security or Network Security rights profile can modify security settings. The Security Administrator role includes these profiles.

To use the LDAP toolbox, you must have completed "Configuring the Solaris Management Console for LDAP (Task Map)" on page 113.

1. **Start the Solaris Management Console.**

   For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

2. **Use the appropriate tool.**

   - To modify a template, use the Security Templates tool.

     All currently defined templates display in the right pane. When you select or create a template, online help is available in the left pane.
   - To assign a host to a template, use the Security Templates tool.
   - To create a host that can be assigned to a template, use the Computers and Networks tool.
   - To assign a label to a zone, use the Trusted Network Zones tool. For more information about zones in Trusted Extensions, see Chapter 16, "Managing Zones in Trusted Extensions (Tasks)."

▼ How to Construct a Remote Host Template

Before You Begin You must be in the global zone in a role that can modify network security. For example, roles that are assigned the Information Security or Network Security rights profiles can modify security settings. The Security Administrator role includes these profiles.

1. **In the Solaris Management Console, navigate to the Security Templates tool.**

   See "How to Open the Trusted Networking Tools" on page 272 for the steps.

2. **Under Computers and Networks, double-click Security Templates.**

   The existing templates are displayed in the View pane. These templates describe the security attributes for hosts that this system can contact. These hosts include CIPSO hosts that are running Trusted Extensions and unlabeled hosts.

3 **Examine the `cipso` template.**

View which hosts and which networks are already assigned this template.

4 **Examine the `admin_low` template.**

View which hosts and which networks are already assigned this template.

5 **Create a template.**

If the provided templates do not sufficiently describe the hosts that can be in communication with this system, choose Add Template from the Action menu.

Use the online help for assistance. Before assigning hosts to the templates, create all the templates that your site requires.

6 **(Optional) Modify an existing template that is not a default template.**

Double-click the template, and use the online help for assistance. You can change the assigned hosts or the assigned networks.

Example 19–1 Creating a Security Template With a Different DOI Value

In this example, the security administrator's network has a DOI whose value is different from `1`. The team that initially configured the system has completed "Configure the Domain of Interpretation" on page 57.

First, the security administrator confirms the value of the DOI in the `/etc/system` file:

```
# grep doi /etc/system
set default_doi = 4
```

Then, in the Security Templates tool, for every template that the administrator creates, the value of `doi` is set to `4`. For the single-label system that is described in Example 19–2, the security administrator creates the following template:

```
template: CIPSO_PUBLIC
host_type: CIPSO
doi: 4
min_sl: PUBLIC
max_sl: PUBLIC
```

Example 19–2 Creating a Security Template That Has a Single Label

In this example, the security administrator wants to create a gateway that can only pass packets at a single label, `PUBLIC`. Using the Security Templates tool in the Solaris Management Console, the administrator creates a template and assigns the gateway host to the template.

First, the gateway host and IP address are added to the Computers and Networks tool.

```
gateway-1
192.168.131.75
```

Then, the template is created in the Security Templates tool. The following are the values in the template:

```
template: CIPSO_PUBLIC
host_type: CIPSO
doi: 1
min_sl: PUBLIC
max_sl: PUBLIC
```

The tool supplies the hexadecimal value for `PUBLIC`, `0X0002-08-08`.

Finally, the `gateway-1` host is assigned to the template by its name and IP address.

```
gateway-1
192.168.131.75
```

On a local host, the `tnrhtp` entry appears similar to the following:

```
cipso_public:host_type=cipso;doi=1;min_sl=0X0002-08-08;max_sl=0X0002-08-08;
```

On a local host, the `tnrhdb` entry appears similar to the following:

```
# gateway-1
192.168.131.75:cipso_public
```

Example 19–3 Creating a Security Template for an Unlabeled Router

Any IP router can forward messages with CIPSO labels even though the router does not explicitly support labels. Such an unlabeled router needs a default label to define the level at which connections to the router, perhaps for router management, need to be handled. In this example, the security administrator creates a router that can forward traffic at any label, but all direct communication with the router is handled at the default label, `PUBLIC`.

In the Solaris Management Console, the administrator creates a template and assigns the gateway host to the template.

First, the router and its IP address are added to the Computers and Networks tool.

```
router-1
192.168.131.82
```

Then, the template is created in the Security Templates tool. The following values are in the template:

```
Template Name: UNL_PUBLIC
Host Type: UNLABELED
DOI: 1
Default Label: PUBLIC
Minimum Label: ADMIN_LOW
Maximum Label: ADMIN_HIGH
```

The tool supplies the hexadecimal value for the labels.

Finally, the `router-1` router is assigned to the template by its name and IP address.

```
router-1
192.168.131.82
```

Example 19–4 Creating a Security Template That Has a Limited Label Range

In this example, the security administrator wants to create a gateway that restricts packets to a narrow label range. In the Solaris Management Console, the administrator creates a template and assigns the gateway host to the template.

First, the host and its IP address are added to the Computers and Networks tool.

```
gateway-ir
192.168.131.78
```

Then, the template is created in the Security Templates tool. The following values are in the template:

```
Template Name: CIPSO_IUO_RSTRCT
Host Type: CIPSO
DOI: 1
Minimum Label: CONFIDENTIAL : INTERNAL USE ONLY
Maximum Label: CONFIDENTIAL : RESTRICTED
```

The tool supplies the hexadecimal value for the labels.

Finally, the `gateway-ir` gateway is assigned to the template by its name and IP address.

```
gateway-ir
192.168.131.78
```

Example 19–5 Creating a Security Template That Has a Security Label Set

In this example, the security administrator wants to create a security template that recognizes two labels only. In the Solaris Management Console, the administrator creates a template and assigns the gateway host to the template.

First, each host and IP address that is going to use this template is added to the Computers and Networks tool.

```
host-slset1
192.168.132.21

host-slset2
192.168.132.22

host-slset3
192.168.132.23

host-slset4
192.168.132.24
```

Then, the template is created in the Security Templates tool. The following values are in the template:

```
Template Name: CIPSO_PUB_RSTRCT
Host Type: CIPSO
DOI: 1
Minimum Label: PUBLIC
Maximum Label: CONFIDENTIAL : RESTRICTED
SL Set: PUBLIC, CONFIDENTIAL : RESTRICTED
```

The tool supplies the hexadecimal value for the labels.

Finally, the range of IP addresses are assigned to the template by using the Wildcard button and a prefix.

```
192.168.132.0/17
```

Example 19–6 Creating an Unlabeled Template at the Label `PUBLIC`

In this example, the security administrator allows a subnetwork of Solaris systems to have the `PUBLIC` label in the trusted network. The template has the following values:

```
Template Name: public
Host Type: Unlabeled
Default Label: Public
Minimum Label: Public
Maximum Label: Public
DOI: 1

Wildcard Entry: 10.10.0.0
Prefix: 16
```

All systems on the `10.10.0.0` subnetwork are handled at the label `PUBLIC`.

Example 19–7 Creating a Labeled Template for Developers

In this example, the security administrator creates a SANDBOX template. This template is assigned to systems that are used by developers of trusted software. The two systems that are assigned this template create and test labeled programs. However, their tests do not affect the other labeled systems, because the label SANDBOX is disjoint from the other labels on the network.

```
Template Name: cipso_sandbox
Host Type: CIPSO
Minimum Label: SANDBOX
Maximum Label: SANDBOX
DOI: 1

Hostname: DevMachine1
IP Address: 196.168.129.129

Hostname: DevMachine2
IP Address: 196.168.129.102
```

The developers who use these systems can communicate with each other at the label SANDBOX.

▼ How to Add Hosts to the System's Known Network

The Computers tool in the Solaris Management Console is identical to the Computers tool in the Solaris OS. This procedure is provided here for your convenience. After the hosts are known, you then assign the hosts to a security template.

Before You Begin You must be in an administrator who can manage networks. For example, roles that include the Network Management or System Administrator rights profiles can manage networks.

1. **In the Solaris Management Console, navigate to the Computers tool.**

   For details, see "How to Open the Trusted Networking Tools" on page 272.

2. **In the Computers tool, confirm that you want to view all computers on the network.**

3. **Add a host that this system can contact.**

   You must add every host that this system might contact, including any static routers and any audit servers.

   a. **From the Action menu, choose Add Computer.**

   b. **Identify the host by name and IP address.**

   c. **(Optional) Provide additional information about the host.**

d. **To add the host, click Apply.**

e. **When the entries are complete, click OK.**

4 **Add a group of hosts that this system can contact.**

Use the online help to add groups of hosts by using a network IP address.

▼ How to Assign a Security Template to a Host or a Group of Hosts

Before You Begin You must be in the Security Administrator role in the global zone.

All hosts that you want to assign to a template must exist in the Computers and Networks tool. For details, see "How to Add Hosts to the System's Known Network" on page 277.

1 **In the Solaris Management Console, navigate to the Security Templates tool.**

For details, see "How to Open the Trusted Networking Tools" on page 272.

2 **Double-click the appropriate template name.**

3 **Click the Hosts Assigned to Template tab.**

4 **To assign the template to a single host, do the following:**

a. **In the Hostname field, type the host's name.**

b. **In the IP Address field, type the host's address.**

c. **Click the Add button.**

d. **To save your changes, click OK.**

5 **To assign a template to a group of hosts with contiguous addresses, do the following:**

a. **Click Wildcard.**

b. **In the IP Address field, type the IP address.**

c. **In the Prefix field, type the prefix that describes the group of contiguous addresses.**

d. **Click the Add button.**

e. **To save your changes, click OK.**

Example 19–8 Adding an IPv4 Network as a Wildcard Entry

In the following example, a security administrator assigns several IPv4 subnetworks to the same security template. In the Hosts Assigned to Template tab, the administrator adds the following wildcard entries:

```
IP Address: 192.168.113.0
IP address: 192.168.75.0
```

Example 19–9 Adding a List of IPv4 Hosts as a Wildcard Entry

In the following example, a security administrator assigns contiguous IPv4 addresses that are not along octet boundaries to the same security template. In the Hosts Assigned to Template tab, the administrator adds the following wildcard entries:

```
IP Address: 192.168.113.100
Prefix Length: 25
```

This wildcard entry covers the address range of `192.168.113.0` to `192.168.113.127`. The address includes `192.168.113.100`.

Example 19–10 Adding a List of IPv6 Hosts as a Wildcard Entry

In the following example, a security administrator assigns contiguous IPv6 addresses to the same security template. In the Hosts Assigned to Template tab, the administrator adds the following wildcard entries:

```
IP Address: 2001:a08:3903:200::0
Prefix Length: 56
```

This wildcard entry covers the address range of `2001:a08:3903:200::0` to `2001:a08:3903:2ff:ffff:ffff:ffff:ffff`. The address includes `2001:a08:3903:201:20e:cff:fe08:58c`.

▼ How to Limit the Hosts That Can Be Contacted on the Trusted Network

This procedure protects labeled hosts from being contacted by arbitrary unlabeled hosts. When Trusted Extensions is installed, this default template defines every host on the network. Use this procedure to enumerate specific unlabeled hosts.

The local `tnrhdb` file on each system is used to contact the network at boot time. By default, every host that is not provided with a CIPSO template is defined by the `admin_low` template. This template assigns every system that is not otherwise defined (`0.0.0.0`) to be an unlabeled system with the default label of `admin_low`.

Caution – The default admin_low template can be a security risk on a Trusted Extensions network. If site security requires strong protection, the security administrator can remove the 0.0.0.0 wildcard entry after the system is installed. The entry must be replaced with entries for every host that the system contacts during boot.

For example, DNS servers, home directory servers, audit servers, broadcast and multicast addresses, and routers must be in the local tnrhdb file after the 0.0.0.0 wildcard entry is removed.

If an application initially recognizes clients at the host address 0.0.0.0, then you must add the 0.0.0.0/32:admin_low host entry to the tnrhdb database. For example, to receive initial connection requests from potential Sun Ray clients, Sun Ray servers must include this entry. Then, when the server recognizes the clients, the clients are provided an IP address and connected as CIPSO clients.

Before You Begin You must be in the Security Administrator role in the global zone.

All hosts that are to be contacted at boot time must exist in the Computers and Networks tool.

1. **In the Solaris Management Console, navigate to the Security Templates tool in the Files scope.**

   The Files scope protects the system during boot. To access the Security Templates tool, see "How to Open the Trusted Networking Tools" on page 272.

2. **Modify the hosts that are assigned to the `admin_low` template.**

   a. **Double-click the `admin_low` template.**

   Every host that is added can be contacted during boot at the label ADMIN_LOW.

   b. **Click the Hosts Assigned to Template tab.**

   Every host that is added can be contacted during boot at the label ADMIN_LOW.

   c. **Add each unlabeled host that must be contacted at boot time.**

   For details, see "How to Assign a Security Template to a Host or a Group of Hosts" on page 278.

   Include every on-link router that is not running Trusted Extensions, through which this host must communicate.

   d. **Add the ranges of hosts that must be contacted at boot time.**

   e. **Remove the `0.0.0.0` entry.**

3 **Modify the hosts that are assigned to the `cipso` template.**

a. **Double-click the `cipso` template.**

Every host that is added can be contacted during boot.

b. **Click the Hosts Assigned to Template tab.**

Every host that is added can be contacted during boot at the label `ADMIN_LOW`.

c. **Add each labeled host that must be contacted at boot time.**

For details, see "How to Assign a Security Template to a Host or a Group of Hosts" on page 278.

- Include the LDAP server.
- Include every on-link router that is running Trusted Extensions, through which this host must communicate
- Make sure that all network interfaces are assigned to the template.
- Include broadcast addresses.

d. **Add the ranges of hosts that must be contacted at boot time.**

4 **Verify that the host assignments allow the system to boot.**

Example 19–11 Changing the Label of the `0.0.0.0` `tnrhdb` Entry

In this example, the security administrator creates a public gateway system. The administrator removes the `0.0.0.0` entry from the `admin_low` template and assigns the entry to an unlabeled template that is named `public`. The system then recognizes any system that is not listed in its `tnrhdb` file as an unlabeled system with the security attributes of the `public` security template.

The following describes an unlabeled template that was created specifically for public gateways.

```
Template Name: public
Host Type: Unlabeled
Default Label: Public
Minimum Label: Public
Maximum Label: Public
DOI: 1
```

Example 19–12 Enumerating Computers to Contact During Boot in the `tnrhdb` Database

The following example shows the local `tnrhdb` database with entries for an LDAP client with two network interfaces. The client communicates with another network and with routers.

```
127.0.0.1:cipso             Loopback address
192.168.112.111:cipso       Interface 1 of this host
192.168.113.111:cipso       Interface 2 of this host
10.6.6.2:cipso              LDAP server
192.168.113.6:cipso         Audit server
192.168.112.255:cipso       Subnet broadcast address
192.168.113.255:cipso       Subnet broadcast address
192.168.113.1:cipso         Router
192.168.117.0:cipso         Another Trusted Extensions network
192.168.112.12:public       Specific network router
192.168.113.12:public       Specific network router
224.0.0.2:public            Multicast address
255.255.255.255:admin_low       Broadcast address
```

Example 19–13 Making the Host Address `0.0.0.0` a Valid `tnrhdb` Entry

In this example, the security administrator configures a Sun Ray server to accept initial connection requests from potential clients. The server is using a private topology and is using the defaults:

```
# utadm -a bge0
```

First, the administrator determines the Solaris Management Console domain name:

```
SMCserver # /usr/sadm/bin/dtsetup scopes
Getting list of managable scopes...
Scope 1 file:/machine1.ExampleCo.COM/machine1.ExampleCo.COM
```

Then, the administrator adds the entry for client initial connection to the Sun Ray server's `tnrhdb` database. Because the administrator is testing, the default wildcard address is still used for all unknown addresses:

```
SunRayServer # /usr/sadm/bin/smtnrhdb \
add -D file:/machine1.ExampleCo.COM/machine1.ExampleCo.COM \
-- -w 0.0.0.0 -p 32 -n admin_low
Authenticating as user: root

Please enter a string value for: password ::
... from machine1.ExampleCo.COM was successful.
```

After this command, the `tnhrdb` database appears similar to the following. The result of the `smtnrhdb` command is highlighted:

```
## tnrhdb database
## Sun Ray server address
       192.168.128.1:cipso
## Sun Ray client addresses on 192.168.128 network
```

```
       192.168.128.0/24:admin_low
## Initial address for new clients
       0.0.0.0/32:admin_low
## Default wildcard address
0.0.0.0:admin_low
```
Other addresses to be contacted at boot

```
# tnchkdb -h /etc/security/tsol/tnrhdb
```

After this phase of testing succeeds, the administrator makes the configuration more secure by removing the default wildcard address, checks the syntax of the `tnrhdb` database, and tests again. The final `tnhrdb` database appears similar to the following:

```
## tnrhdb database
## Sun Ray server address
       192.168.128.1:cipso
## Sun Ray client addresses on 192.168.128 network
       192.168.128.0/24:admin_low
## Initial address for new clients
       0.0.0.0/32:admin_low
## 0.0.0.0:admin_low - no other systems can enter network at admin_low
```
Other addresses to be contacted at boot

Configuring Routes and Checking Network Information in Trusted Extensions (Task Map)

The following task map describes tasks to configure the network and to verify the configuration.

Task	Description	For Instructions
Configure static routes.	Manually describes the best route from one host to another host.	"How to Configure Routes With Security Attributes" on page 284
Check the accuracy of the local network databases.	Uses the `tnchkdb` command to check the syntactic validity of the local network databases.	"How to Check the Syntax of Trusted Network Databases" on page 285
Compare the network database entries with the entries in the kernel cache.	Uses the `tninfo` command to determine if the kernel cache has been updated with the latest database information.	"How to Compare Trusted Network Database Information With the Kernel Cache" on page 286
Synchronize the kernel cache with the network databases.	Uses the `tnctl` command to update the kernel cache with up-to-date network database information on a running system.	"How to Synchronize the Kernel Cache With Trusted Network Databases" on page 287

▼ How to Configure Routes With Security Attributes

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Add every destination host and gateway that you are using to route packets over the trusted network.**

   The addresses are added to the local `/etc/hosts` file, or to its equivalent on the LDAP server. Use the Computers and Networks tool in the Solaris Management Console. The Files scope modifies the `/etc/hosts` file. The LDAP scope modifies the entries on the LDAP server. For details, see "How to Add Hosts to the System's Known Network" on page 277.

2. **Assign each destination host, network, and gateway to a security template.**

   The addresses are added to the local `/etc/security/tsol/tnrhdb` file, or to its equivalent on the LDAP server. Use the Security Templates tool in the Solaris Management Console. For details, see "How to Assign a Security Template to a Host or a Group of Hosts" on page 278.

3. **Set up the routes.**

   In a terminal window, use the `route add` command to specify routes.

   The first entry sets up a default route. The entry specifies a gateway's address, `192.168.113.1`, to use when no specific route is defined for either the host or the packet's destination.

   ```
   # route add default 192.168.113.1  -static
   ```

   For details, see the `route(1M)` man page.

4. **Set up one or more network entries.**

   Use the `-secattr` flag to specify security attributes.

   In the following list of commands, the second line shows a network entry. The third line shows a network entry with a label range of `PUBLIC` to `CONFIDENTIAL : INTERNAL USE ONLY`.

   ```
   # route add default 192.168.113.36
   # route add -net 192.168.102.0 gateway-101
   # route add -net 192.168.101.0 gateway-102 \
   -secattr min_sl="PUBLIC",max_sl="CONFIDENTIAL : INTERNAL USE ONLY",doi=1
   ```

5. **Set up one or more host entries.**

   The new fourth line shows a host entry for the single-label host, `gateway-pub`. `gateway-pub` has a label range of `PUBLIC` to `PUBLIC`.

   ```
   # route add default 192.168.113.36
   # route add -net 192.168.102.0 gateway-101
   # route add -net 192.168.101.0 gateway-102 \
   -secattr min_sl="PUBLIC",max_sl="CONFIDENTIAL : INTERNAL USE ONLY",doi=1
   # route add -host 192.168.101.3 gateway-pub \
   -secattr min_sl="PUBLIC",max_sl="PUBLIC",doi=1
   ```

Example 19–14 Adding a Route With a Label Range of `CONFIDENTIAL : INTERNAL USE ONLY` to `CONFIDENTIAL : RESTRICTED`

The following `route` command adds to the routing table the hosts at `192.168.115.0` with `192.168.118.39` as its gateway. The label range is from `CONFIDENTIAL : INTERNAL USE ONLY` to `CONFIDENTIAL : RESTRICTED`, and the DOI is `1`.

```
$ route add -net 192.168.115.0 192.168.118.39 \
-secattr min_sl="CONFIDENTIAL : INTERNAL USE ONLY",max_sl="CONFIDENTIAL : RESTRICTED",doi=1
```

The result of the added hosts is shown with the `netstat -rR` command. In the following excerpt, the other routes are replaced by ellipses (...).

```
$ netstat -rRn
...
192.168.115.0        192.168.118.39        UG       0      0
        min_sl=CNF : INTERNAL USE ONLY,max_sl=CNF : RESTRICTED,DOI=1,CIPSO
...
```

How to Check the Syntax of Trusted Network Databases

The `tnchkdb` command checks that the syntax of each network database is accurate. The Solaris Management Console runs this command automatically when you use the Security Templates tool or the Trusted Network Zones tool. Typically, you run this command to check the syntax of database files that you are configuring for future use.

Before You Begin You must be in the global zone in a role that can check network settings. The Security Administrator role and the System Administrator role can check these settings.

- **In a terminal window, run the `tnchkdb` command.**

```
$ tnchkdb [-h tnrhdb-path] [-t tnrhtp-path] [-z tnzonecfg-path]
checking /etc/security/tsol/tnrhtp ...
checking /etc/security/tsol/tnrhdb ...
checking /etc/security/tsol/tnzonecfg ...
```

Example 19–15 Testing the Syntax of a Trial Network Database

In this example, the security administrator is testing a network database file for possible use. Initially, the administrator uses the wrong option. The results of the check are printed on the line for the `tnrhdb` file:

```
$ tnchkdb -h /opt/secfiles/trial.tnrhtp
checking /etc/security/tsol/tnrhtp ...
checking /opt/secfiles/trial.tnrhtp ...
```

```
line 12: Illegal name: min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH
line 14: Illegal name: min_sl=ADMIN_LOW;max_sl=ADMIN_HIGH
checking /etc/security/tsol/tnzonecfg ...
```

When the security administrator checks the file by using the `-t` option, the command confirms that the syntax of the trial `tnrhtp` database is accurate:

```
$ tnchkdb -t /opt/secfiles/trial.tnrhtp
checking /opt/secfiles/trial.tnrhtp ...
checking /etc/security/tsol/tnrhdb ...
checking /etc/security/tsol/tnzonecfg ...
```

▼ How to Compare Trusted Network Database Information With the Kernel Cache

The network databases might contain information that is not cached in the kernel. This procedure checks that the information is identical. When you use the Solaris Management Console to update the network, the kernel cache is updated with network database information. The `tninfo` command is useful during testing and for debugging.

Before You Begin You must be in the global zone in a role that can check network settings. The Security Administrator role and the System Administrator role can check these settings.

● **In a terminal window, run the `tninfo` command.**

- `tninfo -h` *hostname* displays the IP address and template for the specified host.
- `tninfo -t` *templatename* displays the following information:

  ```
  template: template-name
  host_type: either CIPSO or UNLABELED
  doi: 1
  min_sl: minimum-label
  hex: minimum-hex-label
  max_sl: maximum-label
  hex:maximum-hex-label
  ```
- `tninfo -m` *zone-name* displays the multilevel port (MLP) configuration of a zone.

Example 19–16 Displaying Multilevel Ports on a Host

In this example, a system is configured with several labeled zones. All zones share the same IP address. Some zones are also configured with zone-specific addresses. In this configuration, the TCP port for web browsing, port `8080`, is an MLP on a shared interface in the public zone. The administrator has also set up `telnet`, TCP port `23`, to be an MLP in the public zone. Because these two MLPs are on a shared interface, no other zone, including the global zone, can receive packets on the shared interface on ports `8080` and `23`.

In addition, the TCP port for `ssh`, port `22`, is a per-zone MLP in the public zone. The public zone's `ssh` service can receive any packets on its zone-specific address within the address's label range.

The following command shows the MLPs for the public zone:

```
$ tninfo -m public
private: 22/tcp
shared:  23/tcp;8080/tcp
```

The following command shows the MLPs for the global zone. Note that ports `23` and `8080` cannot be MLPs in the global zone because the global zone shares the same address with the public zone:

```
$ tninfo -m global
private: 111/tcp;111/udp;514/tcp;515/tcp;631/tcp;2049/tcp;
         6000-6003/tcp;38672/tcp;60770/tcp;
shared:  6000-6003/tcp
```

▼ How to Synchronize the Kernel Cache With Trusted Network Databases

When the kernel has not been updated with trusted network database information, you have several ways to update the kernel cache. The Solaris Management Console runs this command automatically when you use the Security Templates tool or the Trusted Network Zones tool.

Before You Begin You must be in the Security Administrator role in the global zone.

● **To synchronize the kernel cache with network databases, run one of the following commands:**

- **Restart the `tnctl` service.**

  Caution – Do not use this method on systems that obtain their trusted network database information from an LDAP server. The local database information would overwrite the information that is obtained from the LDAP server.

  ```
  $ svcadm restart  svc:/network/tnctl
  ```

  This command reads all information from the local trusted network databases into the kernel.

- **Update the kernel cache for your recently added entries.**

  `$ tnctl -h` *hostname*

This command reads only the information from the chosen option into the kernel. For details about the options, see Example 19–17 and the tnctl(1M) man page.

- **Change the `tnd` polling interval.**

  This does not update the kernel cache. However, you can shorten the polling interval to update the kernel cache more frequently. For details, see the example in the tnd(1M) man page.

- **Refresh the `tnd`.**

  This Service Management Facility (SMF) command triggers an immediate update of the kernel with recent changes to trusted network databases.

  ```
  $ svcadm refresh svc:/network/tnd
  ```

- **Restart the `tnd` by using SMF.**

  ```
  $ svcadm restart svc:/network/tnd
  ```

Caution – Avoid running the `tnd` command to restart the `tnd`. This command can interrupt communications that are currently succeeding.

Example 19–17 Updating the Kernel With Your Latest `tnrhdb` Entries

In this example, the administrator has added three addresses to the local `tnrhdb` database. First, the administrator removed the `0.0.0.0` wildcard entry.

```
$ tnctl -d -h 0.0.0.0:admin_low
```

Then, the administrator views the format of the final three entries in the `/etc/security/tsol/tnrhdb` database:

```
$ tail /etc/security/tsol/tnrhdb
#\:\:0:admin_low
127.0.0.1:cipso
#\:\:1:cipso
192.168.103.5:admin_low
192.168.103.0:cipso
0.0.0.0/32:admin_low
```

Then, the administrator updates the kernel cache:

```
$ tnctl -h 192.168.103.5
tnctl -h 192.168.103.0
tnctl -h 0.0.0.0/32
```

Finally, the administrator verifies that the kernel cache is updated. The output for the first entry is similar to the following:

```
$ tninfo -h 192.168.103.5
IP Address: 192.168.103.5
Template: admin_low
```

Example 19–18 Updating Network Information in the Kernel

In this example, the administrator updates the trusted network with a public print server, and then checks that the kernel settings are correct.

```
$ tnctl -h public-print-server
$ tninfo -h public-print-server
IP Address: 192.168.103.55
Template: PublicOnly
$ tninfo -t PublicOnly
==================================
Remote Host Template Table Entries
----------------------------------
template: PublicOnly
host_type: CIPSO
doi: 1
min_sl: PUBLIC
hex: 0x0002-08-08
max_sl: PUBLIC
hex: 0x0002-08-08
```

Configuring Labeled IPsec (Task Map)

The following task map describes tasks that are used to add labels to IPsec protections.

Task	Description	For Instructions
Use IPsec with Trusted Extensions.	Adds labels to IPsec protections.	"How to Apply IPsec Protections in a Multilevel Trusted Extensions Network" on page 290
Use IPsec with Trusted Extensions across an untrusted network.	Tunnels labeled IPsec packets across an unlabeled network.	"How to Configure a Tunnel Across an Untrusted Network" on page 292

How to Apply IPsec Protections in a Multilevel Trusted Extensions Network

In this procedure, you configure IPsec on two Trusted Extensions systems to handle the following conditions:

- The two systems, `enigma` and `partym`, are multilevel Trusted Extensions systems that are operating in a multilevel network.
- Application data is encrypted and protected against unauthorized change within the network.
- The security label of the data is visible in the form of a CIPSO IP option for use by multilabel routers and security devices on the path between the `enigma` and `partym` systems.
- The security labels that `enigma` and `partym` exchange are protected against unauthorized changes.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Define the `enigma` and `partym` systems' IP addresses as multilevel addresses.**

   Follow the procedures in "Configuring Trusted Network Databases (Task Map)" on page 270. Use a template with a CIPSO host type.

2. **Configure IPsec for the `enigma` and `partym` systems.**

   For the procedure, see "How to Secure Traffic Between Two Systems With IPsec" in *System Administration Guide: IP Services*. Use IKE for key management, as described in the following step.

3. **Add labels to IKE negotiations.**

   Follow the procedure in "How to Configure IKE With Preshared Keys" in *System Administration Guide: IP Services*, then modify the `ike/config` file as follows:

   a. **Add the keywords `label_aware`, `multi_label`, and `wire_label inner` to the `enigma` system's `/etc/inet/ike/config` file.**

      The resulting file appears similar to the following. The label additions are highlighted.

```
### ike/config file on enigma, 192.168.116.16
## Global parameters
#
    ## Phase 1 transform defaults
p1_lifetime_secs 14400
p1_nonce_len 40
#
    ## Use IKE to exchange security labels.
label_aware
```

```
#
      ## Defaults that individual rules can override.
  p1_xform
        { auth_method preshared oakley_group 5 auth_alg sha encr_alg des }
  p2_pfs 2
  #
## The rule to communicate with partym
    # Label must be unique
  { label "enigma-partym"
        local_addr 192.168.116.16
        remote_addr 192.168.13.213
        multi_label
        wire_label inner
        p1_xform
         { auth_method preshared oakley_group 5 auth_alg md5 encr_alg aes }
        p2_pfs 5
  }
```

b. **Add the same keywords to the** `ike/config` **file on the** `partym` **system.**

```
### ike/config file on partym, 192.168.13.213
## Global Parameters
#
    p1_lifetime_secs 14400
p1_nonce_len 40
#
    ## Use IKE to exchange security labels.
label_aware
#
    p1_xform
      { auth_method preshared oakley_group 5 auth_alg sha encr_alg des }
p2_pfs 2
## The rule to communicate with enigma
# Label must be unique
{ label "partym-enigma"
      local_addr 192.168.13.213
      remote_addr 192.168.116.16
      multi_label
      wire_label inner
p1_xform
       { auth_method preshared oakley_group 5 auth_alg md5 encr_alg aes }
    p2_pfs 5
}
```

4 **If AH protection of CIPSO IP options cannot be used on the network, use ESP authentication.**

Use `encr_auth_algs` rather than `auth_algs` in the `/etc/inet/ipsecinit.conf` file to handle authentication. ESP authentication does not cover the IP header and IP options, but will authenticate all information after the ESP header.

```
{laddr enigma raddr partym} ipsec {encr_algs any encr_auth_algs any sa shared}
```

Note – You can also add labels to systems that are protected by certificates. Public key certificates are managed in the global zone on Trusted Extensions systems. Modify the `ike/config` files similarly when completing the procedures in "Configuring IKE With Public Key Certificates" in *System Administration Guide: IP Services*.

▼ How to Configure a Tunnel Across an Untrusted Network

This procedure configures an IPsec tunnel across a public network between two Trusted Extensions VPN gateway systems. The example that is used in this procedure is based on the configuration that is illustrated in "Description of the Network Topology for the IPsec Tasks to Protect a VPN" in *System Administration Guide: IP Services*.

Assume the following modifications to the illustration:

- The net `10` networks are multilevel trusted networks. CIPSO IP option security labels are visible on these LANs.
- The net `192.168` networks are single-label untrusted networks that operate at the `PUBLIC` label. These networks do not support CIPSO IP options.
- Labeled traffic between `enigma` and `partym` is protected against unauthorized changes.

Before You Begin You must be in the Security Administrator role in the global zone.

1 **Follow the procedures in "Configuring Trusted Network Databases (Task Map)" on page 270 to define the following:**

a. **Define net `10.0.0.0/8` IP addresses as multilevel.**

Use a template with a `cipso` host type. Set the label range from `ADMIN_LOW` to `ADMIN_HIGH`.

b. **Define net `192.168.0.0/16` IP addresses as unlabeled at label `PUBLIC`.**

Use a template with an `unlabeled` host type. Set the default label to be `PUBLIC`.

c. **Define `Calif-vpn` and `Euro-vpn` Internet facing addresses `192.168.13.213` and `192.168.116.16` as multilevel.**

Use a template with a `cipso` host type. Set the label range from `ADMIN_LOW` to `ADMIN_HIGH`.

2 Create an IPsec tunnel.

Follow the procedure in "How to Protect a VPN With an IPsec Tunnel in Tunnel Mode Over IPv4" in *System Administration Guide: IP Services*. Use IKE for key management, as described in the following step.

3 Add labels to IKE negotiations.

Follow the procedure in "How to Configure IKE With Preshared Keys" in *System Administration Guide: IP Services*, then modify the `ike/config` file as follows:

a. **Add the keywords `label_aware`, `multi_label`, and `wire_label none PUBLIC` to the `enigma` system's `/etc/inet/ike/config` file.**

The resulting file appears similar to the following. The label additions are highlighted.

```
    ### ike/config file on enigma, 192.168.116.16
## Global parameters
#
    ## Phase 1 transform defaults
p1_lifetime_secs 14400
p1_nonce_len 40
#
    ## Use IKE to exchange security labels.
label_aware
#
    ## Defaults that individual rules can override.
p1_xform
      { auth_method preshared oakley_group 5 auth_alg sha encr_alg des }
p2_pfs 2
#
 ## The rule to communicate with partym
   # Label must be unique
{ label "enigma-partym"
      local_addr 192.168.116.16
      remote_addr 192.168.13.213
      multi_label
      wire_label none PUBLIC
      p1_xform
       { auth_method preshared oakley_group 5 auth_alg md5 encr_alg aes }
      p2_pfs 5
    }
```

b. **Add the same keywords to the `ike/config` file on the `partym` system.**

```
### ike/config file on partym, 192.168.13.213
## Global Parameters
#
    p1_lifetime_secs 14400
p1_nonce_len 40
#
```

```
    ## Use IKE to exchange security labels.
label_aware
#
    p1_xform
      { auth_method preshared oakley_group 5 auth_alg sha encr_alg des }
p2_pfs 2
## The rule to communicate with enigma
# Label must be unique
{ label "partym-enigma"
      local_addr 192.168.13.213
      remote_addr 192.168.116.16
      multi_label
      wire_label none PUBLIC
p1_xform
       { auth_method preshared oakley_group 5 auth_alg md5 encr_alg aes }
    p2_pfs 5
}
```

Note – You can also add labels to systems that are protected by certificates. Modify the `ike/config` files similarly when completing the procedures in "Configuring IKE With Public Key Certificates" in *System Administration Guide: IP Services*.

Troubleshooting the Trusted Network (Task Map)

The following task map describes tasks to debug your network.

Task	Description	For Instructions
Determine why two hosts cannot communicate.	Checks that the interfaces on a single system are up.	"How to Verify That a Host's Interfaces Are Up" on page 294
	Uses debugging tools when two hosts cannot communicate with each other.	"How to Debug the Trusted Extensions Network" on page 295
Determine why an LDAP client cannot reach the LDAP server.	Troubleshoots the loss of connection between an LDAP server and a client.	"How to Debug a Client Connection to the LDAP Server" on page 298

▼ How to Verify That a Host's Interfaces Are Up

Use this procedure if your system does not communicate with other hosts as expected.

Before You Begin You must be in the global zone in a role that can check network settings. The Security Administrator role and the System Administrator role can check these settings.

1 Verify that the system's network interface is up.

The following output shows that the system has two network interfaces, `hme0` and `hme0:3`. Neither interface is up.

```
# ifconfig -a
...
hme0: flags=1000843<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.0.11 netmask ffffff00 broadcast 192.168.0.255
hme0:3 flags=1000843<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.0.12 netmask ffffff00 broadcast 192.168.0.255
```

2 If the interface is not up, bring it up and then verify that it is up.

The following output shows that both interfaces are up.

```
# ifconfig hme0 up
# ifconfig -a
...
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,...
hme0:3 flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,..
```

▼ How to Debug the Trusted Extensions Network

To debug two hosts that should be communicating but are not, you can use Trusted Extensions and Solaris debugging tools. For example, Solaris network debugging commands such as `snoop` and `netstat` are available. For details, see the `snoop`(1M) and `netstat`(1M) man pages. For commands that are specific to Trusted Extensions, see Table 8–2.

- For problems with contacting labeled zones, see “Managing Zones (Task Map)” on page 217.
- For debugging NFS mounts, see “How to Troubleshoot Mount Failures in Trusted Extensions” on page 246.
- For debugging LDAP communications, see “How to Debug a Client Connection to the LDAP Server” on page 298.

Before You Begin You must be in the global zone in a role that can check network settings. The Security Administrator role or the System Administrator role can check these settings.

1 To troubleshoot the `tnd` daemon, change the polling interval and collect debugging information.

For details, see the `tnd`(1M) man page.

2 **Check that the hosts that cannot communicate are using the same naming service.**

a. **On each host, check the `nsswitch.conf` file.**

i. **Check the values for the Trusted Extensions databases in the `nsswitch.conf` file.**

For example, at a site that uses LDAP to administer the network, the entries are similar to the following:

```
# Trusted Extensions
tnrhtp: files ldap
tnrhdb: files ldap
```

ii. **If the values are different, correct the `nsswitch.conf` file.**

b. **Check that the LDAP naming service is configured.**

```
$ ldaplist -l
```

c. **Check that both hosts are in the LDAP naming service.**

```
$ ldaplist -l hosts | grep hostname
```

3 **Check that each host is defined correctly.**

a. **Use the Solaris Management Console to verify the definitions.**

- In the Security Templates tool, check that each host is assigned to a security template that is compatible with the security template of the other host.
- For an an unlabeled system, check that the default label assignment is correct.
- In the Trusted Network Zones tool, check that the multilevel ports (MLPs) are correctly configured.

b. **Use the command line to check that the network information in the kernel is current.**

Check that the assignment in each host's kernel cache matches the assignment on the network, and on the other host.

To get security information for the source, destination, and gateway hosts in the transmission, use the `tninfo` command.

- **Display the IP address and the assigned security template for a given host.**

```
$ tninfo -h hostname
IP Address: IP-address
Template: template-name
```

- **Display a template definition.**

```
$ tninfo -t template-name
template: template-name
```

```
host_type: one of CIPSO or UNLABELED
doi: 1
min_sl: minimum-label
hex: minimum-hex-label
max_sl: maximum-label
hex: maximum-hex-label
```

- **Display the MLPs for a zone.**

  ```
  $ tninfo -m zone-name
  private: ports-that-are-specific-to-this-zone-only
  shared: ports-that-the-zone-shares-with-other-zones
  ```

4 **Fix any incorrect information.**

- To change or check network security information, use the Solaris Management Console tools. For details, see "How to Open the Trusted Networking Tools" on page 272
- To update the kernel cache, restart the `tnctl` service on the host whose information is out of date. Allow some time for this process to complete. Then, refresh the `tnd` service. If the refresh fails, try restarting the `tnd` service. For details, see "How to Synchronize the Kernel Cache With Trusted Network Databases" on page 287.

  Rebooting clears the kernel cache. At boot time, the cache is populated with database information. The `nsswitch.conf` file determines whether local databases or LDAP databases are used to populate the kernel.

5 **Collect transmission information to help you in debugging.**

- **Verify your routing configuration.**

  Use the `get` subcommand to the `route` command.

  ```
  $ route get [ip] -secattr sl=label,doi=integer
  ```

  For details, see the `route(1M)` man page.

- **View the label information in packets.**

  Use the `snoop -v` command.

  The `-v` option displays the details of packet headers, including label information. This command provides a lot of detail, so you might want to restrict the packets that the command examines. For details, see the `snoop(1M)` man page.

- **View the routing table entries and the security attributes on sockets.**

  Use the `-R` option with the `netstat -a|-r` command.

  The `-aR` option displays extended security attributes for sockets. The `-rR` option displays routing table entries. For details, see the `netstat(1M)` man page.

▼ How to Debug a Client Connection to the LDAP Server

Misconfiguration of the client entry on the LDAP server can prevent the client from communicating with the server. Similarly, misconfiguration of files on the client can prevent communication. Check the following entries and files when attempting to debug a client-server communication problem.

Before You Begin You must be in the Security Administrator role in the global zone on the LDAP client.

1. **Check that the remote host template for the LDAP server and for the gateway to the LDAP server are correct.**

   ```
   # tninfo -h LDAP-server
   # route get LDAP-server
   # tninfo -h gateway-to-LDAP-server
   ```

   If a remote host template assignment is incorrect, assign the host to the correct template by using the Security Templates tool in the Solaris Management Console.

2. **Check and correct the `/etc/hosts` file.**

   Your system, the interfaces for the labeled zones on your system, the gateway to the LDAP server, and the LDAP server must be listed in the file. You might have more entries.

   Look for duplicate entries. Remove any entries that are labeled zones on other systems. For example, if `Lserver` is the name of your LDAP server, and `LServer-zones` is the shared interface for the labeled zones, remove `LServer-zones` from `/etc/hosts`.

3. **If you are using DNS, check and correct the entries in the `resolv.conf` file.**

   ```
   # more resolv.conf
   search list of domains
   domain domain-name
   nameserver IP-address

   ...
   nameserver IP-address
   ```

4. **Check that the `tnrhdb` and `tnrhtp` entries in the `nsswitch.conf` file are accurate.**

5. **Check that the client is correctly configured on the server.**

   ```
   # ldaplist -l tnrhdb client-IP-address
   ```

6. **Check that the interfaces for your labeled zones are correctly configured on the LDAP server.**

   ```
   # ldaplist -l tnrhdb client-zone-IP-address
   ```

7 Verify that you can `ping` the LDAP server from all currently running zones.

```
# ldapclient list
...
NS_LDAP_SERVERS= LDAP-server-address
# zlogin zone-name1 ping LDAP-server-address
LDAP-server-address is alive
# zlogin zone-name2 ping LDAP-server-address
LDAP-server-address is alive
...
```

8 Configure LDAP and reboot.

a. For the procedure, see "Make the Global Zone an LDAP Client in Trusted Extensions" on page 63.

b. In every labeled zone, re-establish the zone as a client of the LDAP server.

```
# zlogin zone-name1
# ldapclient init \
-a profileName=profileName \
-a domainName=domain \
-a proxyDN=proxyDN \
-a proxyPassword=password LDAP-Server-IP-Address
# exit
# zlogin zone-name2 ...
```

c. Halt all zones, lock the file systems, and reboot.

If you are using Solaris ZFS, halt the zones and lock the file systems before rebooting. If you are not using ZFS, you can reboot without halting the zones and locking the file systems.

```
# zoneadm list
# zoneadm -z zone-name halt
# lockfs -fa
# reboot
```

CHAPTER 20

Multilevel Mail in Trusted Extensions (Overview)

This chapter covers security and multilevel mailers on systems that are configured with Solaris Trusted Extensions.

Multilevel Mail Service

Trusted Extensions provides multilevel mail for any mail application. When regular users start their mailer, the application opens at the user's current label. If users are operating in a multilevel system, they might want to link or copy their mailer initialization files. For details, see "How to Configure Startup Files for Users in Trusted Extensions" on page 184.

Trusted Extensions Mail Features

In Trusted Extensions, the System Administrator role sets up and administers mail servers according to instructions in the Solaris *System Administration Guide: Advanced Administration* and *System Administration Guide: IP Services*. In addition, the security administrator determines how Trusted Extensions mail features need to be configured.

The following aspects of managing mail are specific to Trusted Extensions:

- The `.mailrc` file is at a user's minimum label.

  Therefore, users who work at multiple labels do not have a `.mailrc` file at the higher labels, unless they copy or link the `.mailrc` file in their minimum-label directory to each higher directory.

  The Security Administrator role or the individual user can add the `.mailrc` file to either `.copy_files` or `.link_files`. For a description of these files, see the `updatehome(1M)` man page. For configuration suggestions, see "`.copy_files` and `.link_files` Files" on page 179.

- Your mail reader can run at every label on a system. Some configuration is required to connect a mail client to the server.

  For example, to use Mozilla mail for multilevel mail requires that you configure a Mozilla mail client at each label to specify the mail server. The mail server could be the same or different for each label, but the server must be specified.
- The Mailing Lists tool in the Solaris Management Console manages mail aliases.

  Depending on the scope of the selected Solaris Management Console toolbox, you can update the local `/etc/aliases` file or the LDAP entry on the Sun Java System Directory Server.
- Trusted Extensions software checks host and user labels before sending or forwarding mail.
  - The software checks that the mail is within the accreditation range of the host. The checks are described in this list and in Chapter 19, "Managing Networks in Trusted Extensions (Tasks)."
  - The software checks that the mail is between the account's clearance and minimum label.
  - Users can read email that is received within their accreditation range. During a session, users can read mail only at their current label.

    To contact regular user by email, an administrative role must send mail from a workspace that is at a label that the user can read. The user's default label is usually a good choice.

CHAPTER 21

Managing Labeled Printing (Tasks)

This chapter describes how to use Solaris Trusted Extensions software to configure labeled printing. It also describes how to configure print jobs without the labeling options.

Labels, Printers, and Printing

Trusted Extensions software uses labels to control printer access. Labels are used to control access to printers and to information about queued print jobs. The software also labels printed output. Body pages are labeled, and mandatory banner and trailer pages are labeled. Banner and trailer pages can also include handling instructions.

The system administrator handles basic printer administration. The security administrator role manages printer security, which includes labels and how the labeled output is handled. The administrators follow basic Solaris printer administration procedures, then they assign labels to the print servers and printers.

Trusted Extensions software supports both single-level and multilevel printing. Multilevel printing is implemented in the global zone only. To use the global zone's print server, a labeled zone must have a host name that is different from the global zone. One way to obtain a distinct host name is to assign an IP address to the labeled zone. The address would be distinct from the global zone's IP address.

Restricting Access to Printers and Print Job Information in Trusted Extensions

Users and roles on a system that is configured with Trusted Extensions software create print jobs at the label of their session. The print jobs can print only on printers that recognize that label. The label must be in the printer's label range.

Users and roles can view print jobs whose label is the same as the label of the session. In the global zone, a role can view jobs whose labels are dominated by the label of the zone.

Printers that are configured with Trusted Extensions software print labels on the printer output. Printers that are managed by unlabeled print servers do not print labels on the printer output. Such printers have the same label as their unlabeled server. For example, a Solaris print server can be assigned an arbitrary label in the `tnrhdb` database of the LDAP naming service. Users can then print jobs at that arbitrary label on the Solaris printer. As with Trusted Extensions printers, those Solaris printers can only accept print jobs from users who are working at the label that has been assigned to the print server.

Labeled Printer Output

Trusted Extensions prints security information on body pages and banner and trailer pages. The information comes from the `label_encodings` file and from the `tsol_separator.ps` file.

The security administrator can do the following to modify defaults that set labels and add handling instructions to printer output:

- Localize or customize the text on the banner and trailer pages
- Specify alternate labels to be printed on body pages or in the various fields of the banner and trailer pages
- Change or omit any of the text or labels

The security administrator can also configure user accounts to use printers that do not print labels on the output. Users can also be authorized to selectively not print banners or labels on printer output.

Labeled Body Pages

By default, the “Protect As” classification is printed at the top and bottom of every body page. The “Protect As” classification is the dominant classification when the classification from the job's label is compared to the `minimum protect as classification`. The `minimum protect as classification` is defined in the `label_encodings` file.

For example, if the user is logged in to an Internal Use Only session, then the user's print jobs are at that label. If the `minimum protect as classification` in the `label_encodings` file is Public, then the Internal Use Only label is printed on the body pages.

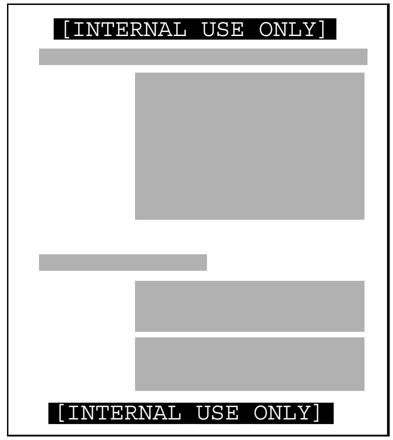


FIGURE 21–1 Job's Label Printed at the Top and Bottom of a Body Page

Labeled Banner and Trailer Pages

The following figures show a default banner page and how the default trailer page differs. Callouts identify the various sections. Note that the trailer page uses a different outer line.

The text, labels, and warnings that appear on print jobs are configurable. The text can also be replaced with text in another language for localization.

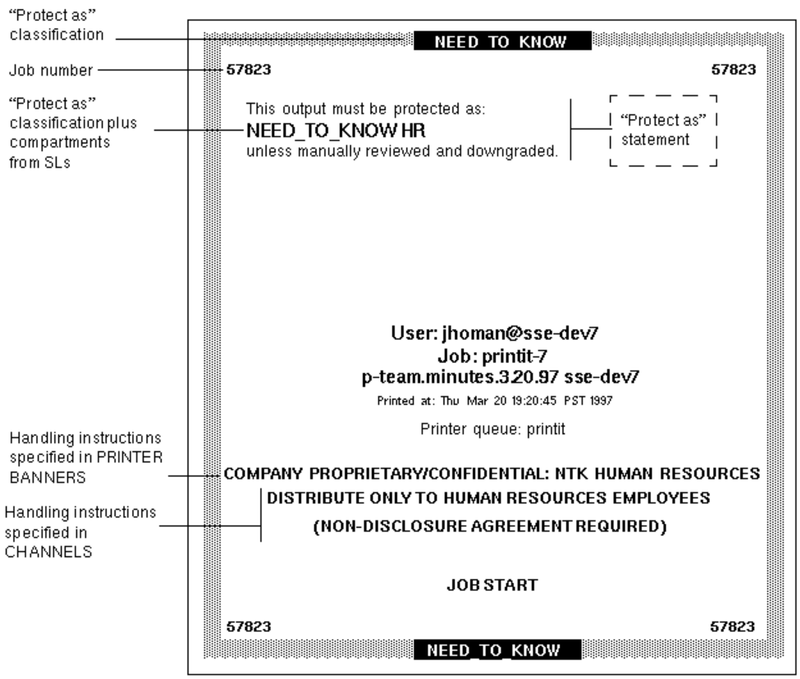


FIGURE 21–2 Typical Banner Page of a Labeled Print Job

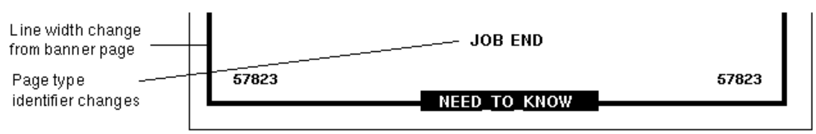


FIGURE 21–3 Differences on a Trailer Page

The following table shows aspects of trusted printing that the security administrator can change by modifying the `/usr/lib/lp/postscript/tsol_separator.ps` file.

Note – To localize or internationalize the printed output, see the comments in the `tsol_separator.ps` file.

TABLE 21–1 Configurable Values in the `tsol_separator.ps` File

Output	Default Value	How Defined	To Change
`PRINTER BANNERS`	`/Caveats Job_Caveats`	`/Caveats Job_Caveats`	See "Specifying Printer Banners" in *Solaris Trusted Extensions Label Administration*.
`CHANNELS`	`/Channels Job_Channels`	`/Channels Job_Channels`	See "Specifying Channels" in *Solaris Trusted Extensions Label Administration*.
Label at the top of banner and trailer pages	`/HeadLabel Job_Protect def`	See `/PageLabel` description.	The same as changing `/PageLabel`.. Also see "Specifying the "Protect As" Classification" in *Solaris Trusted Extensions Label Administration*.
Label at the top and bottom of body pages	`/PageLabel Job_Protect def`	Compares the label of the job to the `minimum protect as classification` in the `label_encodings` file. Prints the more dominant classification. Contains compartments if the print job's label has compartments.	Change the `/PageLabel` definition to specify another value. Or, type a string of your choosing. Or, print nothing at all.
Text and label in the "Protect as" classification statement	`/Protect Job_Protect def`	See `/PageLabel` description.	The same as changing `/PageLabel`.
	`/Protect_Text1 () def`	Text to appear above label.	Replace `()` in `Protect_Text1` and `Protect_Text2` with text string.
	`/Protect_Text2 () def`	Text to appear below label.	

PostScript Printing of Security Information

Labeled printing in Trusted Extensions relies on features from Solaris printing. In the Solaris OS, printer model scripts handle banner page creation. To implement labeling, a printer model script first converts the print job to a PostScript™ file. Then, the PostScript file is manipulated to insert labels on body pages, and to create banner and trailer pages.

Solaris printer model scripts can also translate PostScript into the native language of a printer. If a printer accepts PostScript input, then Solaris software sends the job to the printer. If a printer does not accept PostScript input, then the software converts the PostScript format to a raster image. The raster image is then converted to the appropriate printer format.

Because PostScript software is used to print label information, users cannot print PostScript files by default. This restriction prevents a knowledgeable PostScript programmer from creating a PostScript file that modifies the labels on the printer output.

The Security Administrator role can override this restriction by assigning the `Print Postscript` authorization to role accounts and to trustworthy users. The authorization is assigned only if the account can be trusted not to spoof the labels on printer output. Also, allowing a user to print PostScript files must be consistent with the site's security policy.

Printer Model Scripts

A printer model script enables a particular model of printer to provide banner and trailer pages. Trusted Extensions provides four scripts:

- `tsol_standard` - For directly attached PostScript printers, for example, printers attached by a parallel port
- `tsol_netstandard` - For network–accessible PostScript printers
- `tsol_standard_foomatic` - For directly attached printers that do not print PostScript format
- `tsol_netstandard_foomatic` - For network–accessible printers that do not print PostScript format

The `foomatic` scripts are used when a printer driver name begins with `Foomatic`. Foomatic drivers are PostScript Printer Drivers (PPD). By default, “Use PPD” is specified in the Print Manager when you add a printer. A PPD is then used to translate banner and trailer pages into the language of the printer.

Additional Conversion Filters

A conversion filter converts text files to PostScript format. The filter's programs are trusted programs that are run by the printer daemon. Files that are converted to PostScript format by any installed filter program can be trusted to have authentic labels and banner and trailer page text.

Solaris software provides most conversion filters that a site needs. A site's System Administrator role can install additional filters. These filters can then be trusted to have authentic labels, and banner and trailer pages. To add conversion filters, see Chapter 9, “Customizing LP Printing Services and Printers (Tasks),” in *System Administration Guide: Solaris Printing*.

Interoperability of Trusted Extensions With Trusted Solaris 8 Printing

Trusted Solaris 8 and Trusted Extensions systems that have compatible `label_encodings` files and that identify each other as using a CIPSO template can use each other for remote printing. The following table describes how to set up the systems to enable printing. By default, users cannot list or cancel print jobs on a remote print server of the other OS. Optionally, you can authorize users to do so.

Originating System	Print Server System	Action	Results
Trusted Extensions	Trusted Solaris 8	Configure printing – In the Trusted Extensions `tnrhdb`, assign a template with the appropriate label range to the Trusted Solaris 8 print server. The label could be CIPSO or unlabeled.	Trusted Solaris 8 printer can print jobs from a Trusted Extensions system within the printer's label range.
Trusted Extensions	Trusted Solaris 8	Authorize users – On the Trusted Extensions system, create a profile that adds the needed authorizations. Assign the profile to users.	Trusted Extensions users can list or cancel print jobs that they send to a Trusted Solaris 8 printer. Users cannot view or remove jobs at a different label.
Trusted Solaris 8	Trusted Extensions	Configure printing – In the Trusted Solaris 8 `tnrhdb`, assign a template with the appropriate label range to the Trusted Extensions print server. The label could be CIPSO or unlabeled.	Trusted Extensions printer can print jobs from a Trusted Solaris 8 system within the printer's label range.
Trusted Solaris 8	Trusted Extensions	Authorize users – On the Trusted Solaris 8 system, create a profile that adds the needed authorizations. Assign the profile to users.	Trusted Solaris 8 users can list or cancel print jobs that they send to a Trusted Extensions printer. Users cannot view or remove jobs at a different label.

Trusted Extensions Print Interfaces (Reference)

The following user commands are extended to conform with Trusted Extensions security policy:

- `cancel` – The caller must be equal to the label of the print job to cancel a job. By default, regular users can cancel only their own jobs.
- `lp` – Trusted Extensions adds the `-o nolabels` option. Users must be authorized to print with no labels. Similarly, users must be authorized to use the `-o nobanner` option.
- `lpstat` – The caller must be equal to the label of the print job to obtain the status of a job. By default, regular users can view only their own print jobs.

The following administrative commands are extended to conform with Trusted Extensions security policy. As in the Solaris OS, these commands can only be run by a role that includes the Printer Management rights profile.

- `lpmove` – The caller must be equal to the label of the print job to move a job. By default, regular users can move only their own print jobs.
- `lpadmin` – In the global zone, this command works for all jobs. In a labeled zone, the caller must dominate the print job's label to view a job, and be equal to change a job.

  Trusted Extensions adds printer model scripts to the `-m` option. Trusted Extensions adds the `-o nolabels` option.
- `lpsched` – In the global zone, this command is always successful. As in the Solaris OS, use the `svcadm` command to enable, disable, start, or restart the print service. In a labeled zone, the caller must be equal to the label of the print service to change the print service. For details about the service management facility, see the `smf(5)`, `svcadm(1M)`, and `svcs(1)` man pages.

Trusted Extensions adds the `solaris.label.print` authorization to the Printer Management rights profile. The `solaris.print.unlabeled` authorization is required to print body pages without labels.

Managing Printing in Trusted Extensions (Task Map)

Trusted Extensions procedures for configuring printing are performed after completing Solaris printer setup. The following task map points to the major tasks that manage labeled printing.

Task	Description	For Instructions
Configure printers for labeled output.	Enables users to print to a Trusted Extensions printer. The print jobs are marked with labels.	"Configuring Labeled Printing (Task Map)" on page 310
Remove visible labels from printer output.	Enables users to print at a specific label to a Solaris printer. The print jobs are not marked with labels. Or, prevents labels from printing on a Trusted Extensions printer.	"Reducing Printing Restrictions in Trusted Extensions (Task Map)" on page 316

Configuring Labeled Printing (Task Map)

The following task map describes common configuration procedures that are related to labeled printing.

Note – Printer clients can only print jobs within the label range of the Trusted Extensions print server.

Task	Description	For Instructions
Start the Print Manager.	Uses a GUI to identify the printer to the network or to the local system. The system administrator starts the GUI in an administrative role workspace.	Chapter 6, "Setting Up and Administering Printers by Using LP Print Commands (Tasks)," in *System Administration Guide: Solaris Printing*
Configure printing from the global zone.	Creates a multilevel print server in the global zone.	"How to Configure a Multilevel Print Server and Its Printers" on page 311
Configure printing from a labeled zone.	Creates a single–label print server for a labeled zone.	"How to Configure a Zone for Single-Label Printing" on page 313
Configure a multilevel print client.	Connects a Trusted Extensions host to a printer.	"How to Enable a Trusted Extensions Client to Access a Printer" on page 314
Restrict the label range of a printer.	Limits a Trusted Extensions printer to a narrow label range.	"How to Configure a Restricted Label Range for a Printer" on page 316

▼ How to Configure a Multilevel Print Server and Its Printers

Printers that are managed by a Trusted Extensions print server print labels on body pages, banner pages, and trailer pages. Such printers can print jobs within the label range of the print server. Any Trusted Extensions host that can reach the print server can use the printers that are connected to that server.

Before You Begin Determine the print server for your Trusted Extensions network. You must be in the System Administrator role in the global zone on this print server.

1. **Start the Solaris Management Console.**

   For details, see "How to Administer the Local System With the Solaris Management Console" on page 154.

2. **Choose the Files toolbox.**

   The title of the toolbox includes `Scope=Files, Policy=TSOL`.

3 Enable multilevel printing by configuring the global zone with the print server port, 515/tcp.

Create a multilevel port (MLP) for the print server by adding the port to the global zone.

a. Navigate to the Trusted Network Zones tool.

b. In the Multilevel Ports for Zone's IP Addresses, add 515/tcp.

c. Click OK.

4 Define the characteristics of the connected printers.

a. Start the Print Manager.

b. Define the make and model of a connected printer.

In the Print Manager, you supply the values for the first two fields, then the Print Manager supplies the driver name.

`Printer Make`	*manufacturer*
`Printer Model`	*manufacturer-part-number*
`Printer Driver`	*automatically filled in*

5 Assign a printer model script to each printer that is connected to the print server.

The model script activates the banner and trailer pages for the specified printer.

For your choice of scripts, see "Printer Model Scripts" on page 308. If the driver name for the printer starts with `Foomatic`, then specify one of the foomatic model scripts. Use the following command:

```
$ lpadmin -p printer -m model
```

If the default printer label range of `ADMIN_LOW` to `ADMIN_HIGH` is acceptable for every printer, then your label configuration is done.

See Also

- **Limit printer label range –** "How to Configure a Restricted Label Range for a Printer" on page 316
- **Prevent labeled output –** "Reducing Printing Restrictions in Trusted Extensions (Task Map)" on page 316
- **Use this zone as a print server –** "How to Enable a Trusted Extensions Client to Access a Printer" on page 314
- **Finish printer setup –** Chapter 6, "Setting Up and Administering Printers by Using LP Print Commands (Tasks)," in *System Administration Guide: Solaris Printing*

▼ How to Configure a Zone for Single-Label Printing

Before You Begin The zone must not be sharing an IP address with the global zone. You must be in the System Administrator role in the global zone.

1. **Add a workspace.**

   For details, see "How to Add a Workspace at a Particular Label" in *Solaris Trusted Extensions User's Guide*.

2. **Change the label of the new workspace to the label of the zone that will be the print server for that label.**

   For details, see "How to Change the Label of a Workspace" in *Solaris Trusted Extensions User's Guide*.

3. **Define the characteristics of the connected printers.**

   a. **At the label of zone, start the Print Manager.**

      By default, the "Use PPD" checkbox is selected. The system finds the appropriate driver for the printer.

   b. **(Optional) To specify a different printer driver, do the following:**

      i. **Remove the check from "Use PPD".**

      ii. **Define the make and model of the printer that uses a different driver.**

         In the Print Manager, you supply the values for the first two fields, then the Print Manager supplies the driver name.

         `Printer Make` *manufacturer*
         `Printer Model` *manufacturer-part-number*
         `Printer Driver` *automatically filled in*

4. **Assign a printer model script to each printer that is connected to the zone.**

   The model script activates the banner and trailer pages for the specified printer.

   For your choices of scripts, see "Printer Model Scripts" on page 308. If the driver name for the printer starts with `Foomatic`, then specify one of the foomatic model scripts. Use the following command:

   `$ lpadmin -p` *printer* `-m` *model*

   The attached printers can print jobs only at the label of the zone.

See Also

- **Prevent labeled output –** "Reducing Printing Restrictions in Trusted Extensions (Task Map)" on page 316

- **Use this zone as a print server** – "How to Enable a Trusted Extensions Client to Access a Printer" on page 314
- **Finish printer setup** – Chapter 6, "Setting Up and Administering Printers by Using LP Print Commands (Tasks)," in *System Administration Guide: Solaris Printing*

▼ How to Enable a Trusted Extensions Client to Access a Printer

Initially, only the zone in which a print server was configured can print to the printers of that print server. The system administrator must explicitly add access to those printers for other zones and systems. The possibilities are as follows:

- For a global zone, add access to the printers that are connected to a global zone on a different system.
- For a labeled zone, add access to the printers that are connected to the global zone of its system.
- For a labeled zone, add access to a printer that a remote zone at the same label is configured for.
- For a labeled zone, add access to the printers that are connected to a global zone on a different system.

Before You Begin A print server has been configured with a label range or a single label, and the printers that are connected to it have been configured. For details, see the following:

- "How to Configure a Multilevel Print Server and Its Printers" on page 311
- "How to Configure a Zone for Single-Label Printing" on page 313
- "How to Assign a Label to an Unlabeled Print Server" on page 318

You must be in the System Administrator role in the global zone, or be able to assume the role.

● **Complete the procedures that enable your systems to access a printer.**

To use the Print Manager instead of the `lpadmin` command, see Example 21–1.

- **Configure the global zone on a system that is not a print server to use another system's global zone for printer access.**

  a. **On the system that does not have printer access, assume the System Administrator role.**

  b. **Add access to the printer that is connected to the Trusted Extensions print server.**

  ```
  $ lpadmin -s printer
  ```

- **Configure a labeled zone to use its global zone for printer access.**

  a. **Change the label of the role workspace to the label of the labeled zone.**

     For details, see "How to Change the Label of a Workspace" in *Solaris Trusted Extensions User's Guide*.

  b. **Add access to the printer.**

     ```
     $ lpadmin -s printer
     ```

- **Configure a labeled zone to use another system's labeled zone for printer access.**

  The labels of the zones must be identical.

  a. **On the system that does not have printer access, assume the System Administrator role.**

  b. **Change the label of the role workspace to the label of the labeled zone.**

  c. **Add access to the printer that is connected to the print server of the remote labeled zone.**

     ```
     lpadmin -s printer
     ```

- **Configure a labeled zone to use an unlabeled print server for printer access.**

  The label of the zone must be identical to the label of the print server.

  a. **On the system that does not have printer access, assume the System Administrator role.**

  b. **Change the label of the role workspace to the label of the labeled zone.**

     For details, see "How to Change the Label of a Workspace" in *Solaris Trusted Extensions User's Guide*.

  c. **Add access to the printer that is connected to the arbitrarily labeled print server.**

     ```
     $ lpadmin -s printer
     ```

Example 21–1 Using the Print Manager to Enable Printer Access

Rather than run the `lpadmin` command, choose the Add button from the Print Manager. The Print Manager must be started in the same zone at the same label as the `lpadmin -s` *printer* command.

▼ How to Configure a Restricted Label Range for a Printer

The default printer label range is `ADMIN_LOW` to `ADMIN_HIGH`. This procedure narrows the label range for a printer that is controlled by a Trusted Extensions print server.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Start the Device Manager.**

   Choose the Allocate Device option from the Trusted Path menu.

2. **Click the Administration button to display the Device Administration dialog box.**

3. **Type a name for the new printer.**

   If the printer is attached to your system, find the name of the printer.

4. **Click the Configure button to display the Device Configuration dialog box.**

5. **Change the printer's label range.**

   a. **Click the Min Label button to change the minimum label.**

      Choose a label from the label builder. For information about the label builder, see "Label Builder in Trusted Extensions" on page 145.

   b. **Click the Max Label button to change the maximum label.**

6. **Save the changes.**

   a. **Click OK in the Configuration dialog box.**

   b. **Click OK in the Administration dialog box.**

7. **Close the Device Manager.**

Reducing Printing Restrictions in Trusted Extensions (Task Map)

The following tasks are optional. They reduce the printing security that Trusted Extensions provides by default when the software is installed.

Task	Description	For Instructions
Configure a printer to not label output.	Prevents security information from printing on body pages, and removes banner and trailer pages.	"How to Remove Labels From Printed Output" on page 317
Configure printers at a single label without labeled output.	Enables users to print at a specific label to a Solaris printer. The print jobs are not marked with labels.	"How to Assign a Label to an Unlabeled Print Server" on page 318
Remove visible labeling of body pages.	Modifies the `tsol_separator.ps` file to prevent labeled body pages on all print jobs that are sent from a Trusted Extensions host.	"How to Remove Page Labels From All Print Jobs" on page 319
Suppress banner and trailer pages.	Authorizes specific users to print jobs without banner and trailer pages.	"How to Suppress Banner and Trailer Pages for Specific Users" on page 320
Enable trusted users to print jobs without labels.	Authorizes specific users or all users of a particular system to print jobs without labels.	"How to Enable Specific Users to Suppress Page Labels" on page 319
Enable the printing of PostScript files.	Authorizes specific users or all users of a particular system to print PostScript files.	"How to Enable Users to Print PostScript Files in Trusted Extensions" on page 320
Assign printing authorizations.	Enables users to bypass default printing restrictions.	"How to Create a Rights Profile for Convenient Authorizations" on page 190 "How to Modify `policy.conf` Defaults" on page 182

▼ How to Remove Labels From Printed Output

Printers that do not have a Trusted Extensions printer model script do not print labeled banner or trailer pages. The body pages also do not include labels.

Before You Begin You must be in the Security Administrator role in the global zone.

● **At the appropriate label, do one of the following:**

- **From the print server, stop banner printing altogether.**

  ```
  % lpadmin -p printer -o nobanner=never
  ```

  Body pages are still labeled.

- **Set the printer model script to a Solaris script.**

  ```
  % lpadmin -p printer  \
  -m { standard | netstandard | standard_foomatic | netstandard_foomatic }
  ```

  No labels appear on printed output.

▼ How to Assign a Label to an Unlabeled Print Server

A Solaris print server is an unlabeled print server that can be assigned a label for Trusted Extensions access to the printer at that label. Printers that are connected to an unlabeled print server can print jobs only at the label that has been assigned to the print server. Jobs print without labels or trailer pages and might print without banner pages. If a job prints with a banner page, the page does not contain any security information.

A Trusted Extensions system can be configured to submit jobs to a printer that is managed by an unlabeled print server. Users can print jobs on the unlabeled printer at the label that the security administrator assigns to the print server.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Open the Solaris Management Console in the appropriate scope.**

   For details, see "Initialize the Solaris Management Console Server in Trusted Extensions" on page 60.

2. **Under System Configuration, navigate to the Computers and Networks tool.**

   Provide a password when prompted.

3. **Assign an unlabeled template to the print server.**

   For details, see "How to Assign a Security Template to a Host or a Group of Hosts" on page 278.

   Choose a label. Users who are working at that label can send print jobs to the Solaris printer at the label of the print server. Pages do not print with labels, and banner and trailer pages are also not part of the print job.

Example 21–2 Sending Public Print Jobs to an Unlabeled Printer

Files that are available to the general public are suitable for printing to an unlabeled printer. In this example, marketing writers need to produce documents that do not have labels printed on the top and bottom of the pages.

The security administrator assigns an unlabeled host type template to the Solaris print server. The template is described in Example 19–6. The arbitrary label of the template is `PUBLIC`. The printer `pr-nolabel1` is connected to this print server. Print jobs from users in a `PUBLIC` zone print on the `pr-nolabel1` printer with no labels. Depending on the settings for the printer, the jobs might or might not have banner pages. The banner pages do not contain security information.

▼ How to Remove Page Labels From All Print Jobs

This procedure prevents all print jobs on a Trusted Extensions printer from including visible labels on the body pages of the print job.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Edit the `/usr/lib/lp/postscript/tsol_separator.ps` file.**

   Use the trusted editor. For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

2. **Find the definition of `/PageLabel`.**

   Find the following lines:

```
%% To eliminate page labels completely, change this line to
%% set the page label to an empty string: /PageLabel () def
/PageLabel Job_PageLabel def
```

Note – The value `Job_PageLabel` might be different at your site.

---

3. **Replace the value of `/PageLabel` with a set of empty parentheses.**

```
/PageLabel () def
```

▼ How to Enable Specific Users to Suppress Page Labels

This procedure enables an authorized user or role to print jobs on a Trusted Extensions printer without labels on the top and bottom of each body page. Page labels are suppressed for all labels at which the user can work.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Determine who is permitted to print jobs without page labels.**

2. **Authorize those users and roles to print jobs without page labels.**

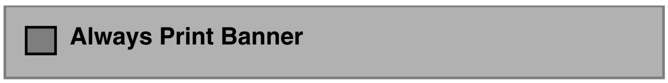
   Assign a rights profile that includes the Print without Label authorization to those users and roles. For details, see "How to Create a Rights Profile for Convenient Authorizations" on page 190.

3. **Instruct the user or role to use the `lp` command to submit print jobs:**

```
% lp -o nolabels staff.mtg.notes
```

How to Suppress Banner and Trailer Pages for Specific Users

Before You Begin The Always Print Banner checkbox in the Print Manager dialog box does not contain a checkmark.



You must be in the Security Administrator role in the global zone.

1. **Create a rights profile that includes the Print without Banner authorization.**

   Assign the profile to each user or role that is allowed to print without banner and trailer pages.

   For details, see "How to Create a Rights Profile for Convenient Authorizations" on page 190.

2. **Instruct the user or role to use the `lp` command to submit print jobs:**

   ```
   % lp -o nobanner staff.mtg.notes
   ```

How to Enable Users to Print PostScript Files in Trusted Extensions

Before You Begin You must be in the Security Administrator role in the global zone.

- **Use one of the following three methods to enable users to print PostScript files:**
  - **To enable PostScript printing on a system, modify the `/etc/default/print` file.**

    a. **Create or modify the `/etc/default/print` file.**

       Use the trusted editor. For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

    b. **Type the following entry:**

       ```
       PRINT_POSTSCRIPT=1
       ```

    c. **Save the file and close the editor.**

- **To authorize all users to print PostScript files from a system, modify the `/etc/security/policy.conf` file.**

  a. **Modify the `policy.conf` file.**

     Use the trusted editor. For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

  b. **Add the `solaris.print.ps` authorization.**

     AUTHS_GRANTED=*other-authorizations*,solaris.print.ps

  c. **Save the file and close the editor.**

- **To enable a user or role to print PostScript files from any system, give just those users and roles the appropriate authorization.**

  Assign a profile that includes the `Print Postscript` authorization to those users and roles. For details, see "How to Create a Rights Profile for Convenient Authorizations" on page 190.

Example 21–3 Enabling PostScript Printing From a Public System

In the following example, the security administrator has constrained a public kiosk to operate at the `PUBLIC` label. The system also has a few icons that open topics of interest. These topics can be printed.

The security administrator creates an `/etc/default/print` file on the system. The file has one entry to enable the printing of PostScript files. No user needs a `Print Postscript` authorization.

```
# vi /etc/default/print

# PRINT_POSTSCRIPT=0
PRINT_POSTSCRIPT=1
```

CHAPTER 22

Devices in Trusted Extensions (Overview)

This chapter describes the extensions that Solaris Trusted Extensions provides to Solaris device protection.

Device Protection With Trusted Extensions Software

On a Solaris system, devices can be protected by allocation and by authorization. By default, devices are available to regular users without an authorization. A system that is configured with Trusted Extensions software uses the device protection mechanisms of the Solaris OS.

However, by default, Trusted Extensions requires that a device be allocated for use, and that the user be authorized to use the device. In addition, devices are protected by labels. Trusted Extensions provides a graphical user interface (GUI) for administrators to manage devices. The same interface is used by users to allocate devices.

Note – In Trusted Extensions, users cannot use the `allocate` and `deallocate` commands. Users must use the Device Allocation Manager. In Solaris Trusted Extensions (JDS), the title of the GUI is Device Manager.

For information about device protection in the Solaris OS, see Chapter 5, "Controlling Access to Devices (Tasks)," in *System Administration Guide: Security Services*.

On a system that is configured with Trusted Extensions, two roles protect devices.

- The System Administrator role controls access to peripheral devices.

  The system administrator makes a device allocatable. Devices that the system administrator makes nonallocatable cannot be used by anyone. Allocatable devices can be allocated only by authorized users.
- The Security Administrator role restricts the labels at which a device can be accessed and sets device policy. The security administrator decides who is authorized to allocate a device.

The following are the main features of device control with Trusted Extensions software:

- By default, an unauthorized user on a Trusted Extensions system cannot allocate devices such as tape drives, CD-ROM drives, or diskette drives.

  A regular user with the Allocate Device authorization can import or export information at the label at which the user allocates the device.
- Users invoke the Device Allocation Manager to allocate devices when they are logged in directly. To allocate a device remotely, users must have access to the global zone. Typically, only roles have access to the global zone.
- The label range of each device can be restricted by the security administrator. Regular users are limited to accessing devices whose label range includes the labels at which the users are allowed to work. The default label range of a device is `ADMIN_LOW` to `ADMIN_HIGH`.
- Label ranges can be restricted for both allocatable and nonallocatable devices. Nonallocatable devices are devices such as frame buffers and printers.

Device Label Ranges

To prevent users from copying sensitive information, each allocatable device has a label range. To use an allocatable device, the user must be currently operating at a label within the device's label range. If the user is not, allocation is denied. The user's current label is applied to data that is imported or exported while the device is allocated to the user. The label of exported data is displayed when the device is deallocated. The user must physically label the medium that contains the exported data.

Effects of Label Range on a Device

To restrict direct login access through the console, the security administrator can set a restricted label range on the frame buffer.

For example, a restricted label range might be specified to limit access to a publicly accessible system. The label range enables users to access the system only at a label within the frame buffer's label range.

When a host has a local printer, a restricted label range on the printer limits the jobs that can be printed on the printer.

Device Access Policies

Trusted Extensions follows the same device policies as the Solaris OS. The security administrator can change default policies and define new policies. The `getdevpolicy` command retrieves information about device policy, and the `update_drv` command changes device policy. For more information, see "Configuring Device Policy (Task Map)" in *System Administration Guide: Security Services*. See also the `getdevpolicy(1M)` and `update_drv(1M)` man pages.

Device-Clean Scripts

A device-clean script is run when a device is allocated or deallocated. The Solaris OS provides scripts for tape drives, CD-ROM drives, and diskette drives. If your site adds allocatable device types to the system, the added devices might need scripts. To see existing scripts, go to the `/etc/security/lib` directory. For more information, see "Device-Clean Scripts" in *System Administration Guide: Security Services*.

For Trusted Extensions software, device-clean scripts must satisfy certain requirements. These requirements are described in the `device_clean(5)` man page.

Device Manager GUI

The Device Manager is used by administrators to administer allocatable and nonallocatable devices. The Device Manager is also used by regular users to allocate and deallocate devices. The users must have the Allocate Device authorization.

The GUI is called the Device Manager. This GUI is started from the Trusted Path menu by selecting Allocate Device. The following figure shows a Device Manager that was opened by a user who can allocate the `audio` device.

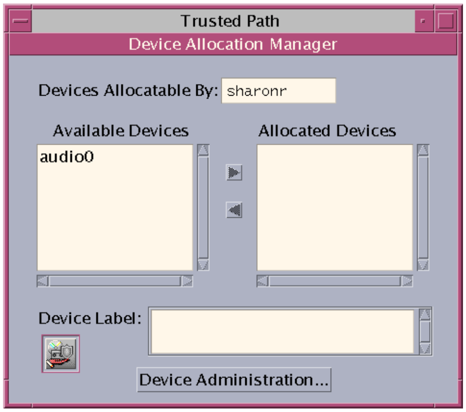
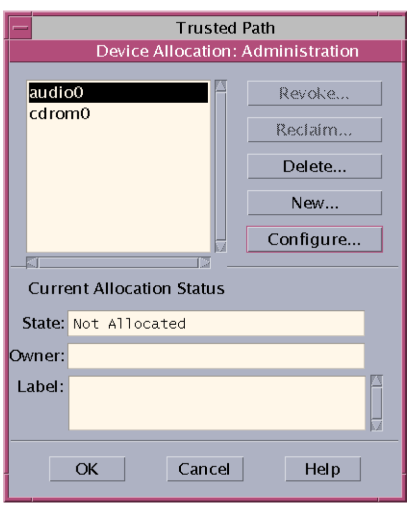


FIGURE 22–1 Device Manager Opened by a User

Users see an empty list when they are not authorized to allocate devices. Or, an empty list might indicate that the allocatable devices are currently allocated by another user or are in an error state. If a user cannot see a device in the Available Devices list, the user needs to contact the responsible administrator.

The Device Administration feature is available to roles that have either one or both of the authorizations that are needed to administer devices. The administration authorizations are Configure Device Attributes, and Revoke or Reclaim Device. The following figure shows a Device Allocation Administration dialog box.



Enforcement of Device Security in Trusted Extensions

The security administrator decides who can allocate devices and makes sure that any user who is authorized to use devices is trained. The user is trusted to do the following:

- Properly label and handle any media containing exported sensitive information so that the information does not become available to anyone who should not see it.

  For example, if information at a label of `NEED TO KNOW ENGINEERING` is stored on a diskette, the person who exports the information must physically label the disk with the `NEED TO KNOW ENGINEERING` label. The diskette must be stored where it is accessible only to members of the engineering group with a need to know.
- Ensure that labels are properly maintained on any information being imported (read) from media on these devices.

  An authorized user must allocate the device at the label that matches the label of the information that is being imported. For example, if a user allocates a diskette drive at `PUBLIC`, the user must only import information labeled `PUBLIC`.

The security administrator is also responsible for enforcing proper compliance with these security requirements.

Devices in Trusted Extensions (Reference)

Trusted Extensions device protection uses Solaris interfaces and Trusted Extensions interfaces.

For Solaris command-line interfaces, see "Device Protection (Reference)" in *System Administration Guide: Security Services*.

Administrators who do not have access to the Device Allocation Manager can administer allocatable devices by using the command line. The `allocate` and `deallocate` commands have administrative options. For examples, see "Forcibly Allocating a Device" in *System Administration Guide: Security Services* and "Forcibly Deallocating a Device" in *System Administration Guide: Security Services*.

For Trusted Extensions command-line interfaces, see the `add_allocatable(1M)` and `remove_allocatable(1M)` man pages.

CHAPTER 23

Managing Devices for Trusted Extensions (Tasks)

This chapter describes how to administer and use devices on a system that is configured with Solaris Trusted Extensions.

Handling Devices in Trusted Extensions (Task Map)

The following task map points to task maps for administrators and users for handling peripheral devices.

Task	Description	For Instructions
Use devices.	Uses a device as a role or as a regular user.	"Using Devices in Trusted Extensions (Task Map)" on page 330
Administer devices.	Configures devices for ordinary users.	"Managing Devices in Trusted Extensions (Task Map)" on page 330
Customize device authorizations.	The Security Administrator role creates new authorizations, adds them to the device, places them in a rights profile and assigns this profile to the user.	"Customizing Device Authorizations in Trusted Extensions (Task Map)" on page 337

Using Devices in Trusted Extensions (Task Map)

In Trusted Extensions, all roles are authorized to allocate a device. Like users, roles must use the Device Manager. The Solaris `allocate` command does not work in Trusted Extensions. The following task map points to user procedures that include using devices to perform administrative tasks.

Task	For Instructions
Allocate and deallocate a device.	"How to Allocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*
Use portable media to transfer files.	"How to Copy Files From Portable Media in Trusted Extensions" on page 99 "How to Copy Files to Portable Media in Trusted Extensions" on page 98

Managing Devices in Trusted Extensions (Task Map)

The following task map describes procedures to protect devices at your site.

Task	Description	For Instructions
Set or modify device policy.	Changes the privileges that are required to access a device.	"Configuring Device Policy (Task Map)" in *System Administration Guide: Security Services*
Authorize users to allocate a device.	The Security Administrator role assigns a profile with the Allocate Device authorization to the user.	"How to Authorize Users to Allocate a Device" in *System Administration Guide: Security Services*
	The Security Administrator role assigns a profile with the site-specific authorizations to the user.	"Customizing Device Authorizations in Trusted Extensions (Task Map)" on page 337
Configure a device.	Chooses security features to protect the device.	"How to Configure a Device in Trusted Extensions" on page 331
Revoke or reclaim a device.	Uses the Device Manager to make a device available for use.	"How to Revoke or Reclaim a Device in Trusted Extensions" on page 333
	Uses Solaris commands to make a device available or unavailable for use.	"Forcibly Allocating a Device" in *System Administration Guide: Security Services* "Forcibly Deallocating a Device" in *System Administration Guide: Security Services*
Prevent access to an allocatable device.	Provides fine–grained access control to a device.	Example 23–3
	Denies everyone access to an allocatable device.	Example 23–1

Task	Description	For Instructions
Protect printers and frame buffers.	Ensures that nonallocatable devices are not allocatable.	"How to Protect Nonallocatable Devices in Trusted Extensions" on page 334
Configure serial login devices.	Enables logins by serial port.	"How to Configure a Serial Line for Logins" on page 335
Use a new device-clean script.	Places a new script in the appropriate places.	"How to Add a Device_Clean Script in Trusted Extensions" on page 336

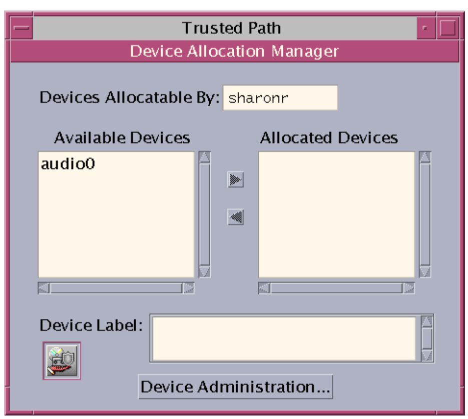
▼ How to Configure a Device in Trusted Extensions

By default, an allocatable device has a label range from `ADMIN_LOW` to `ADMIN_HIGH` and must be allocated for use. Also, users must be authorized to allocate the device. These defaults can be changed.

Before You Begin You must be in the Security Administrator role in the global zone.

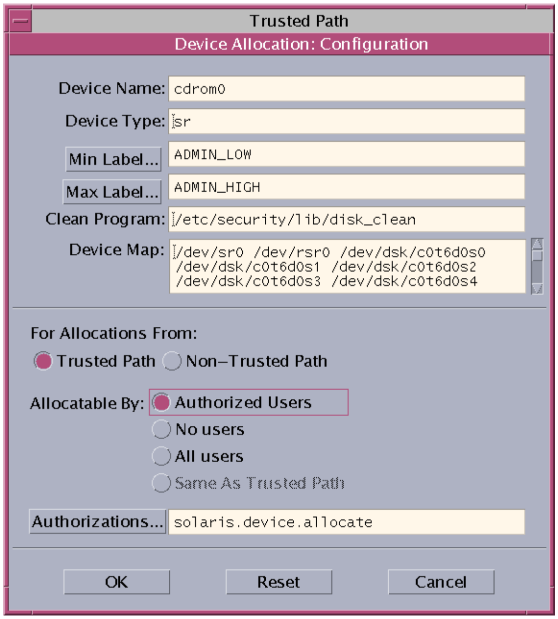
1 From the Trusted Path menu, select Allocate Device.

The Device Manager appears.



2 View the default security settings.

Click Device Administration, then highlight the device. The following figure shows a CD-ROM drive with default security settings.



3 **(Optional) Restrict the label range on the device.**

a. Set the minimum label.

Click the Min Label... button. Choose a minimum label from the label builder. For information about the label builder, see "Label Builder in Trusted Extensions" on page 145.

b. Set the maximum label.

Click the Max Label... button. Choose a maximum label from the label builder.

4 **Specify if the device can be allocated locally.**

In the Device Configuration dialog box, under For Allocations From Trusted Path, select an option from the Allocatable By list. By default, the Authorized Users option is checked. Therefore, the device is allocatable and users must be authorized.

- **To make the device nonallocatable, click No Users.**

  When configuring a printer, frame buffer, or other device that must not be allocatable, select No Users.

- **To make the device allocatable, but to not require authorization, click All Users.**

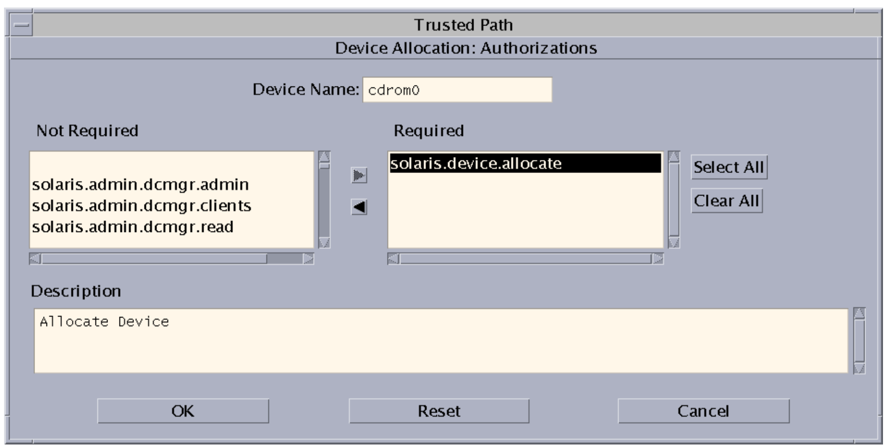
5 Specify if the device can be allocated remotely.

In the For Allocations From Non-Trusted Path section, select an option from the Allocatable By list. By default, the Same As Trusted Path option is checked.

- **To require user authorization, select Allocatable by Authorized Users.**
- **To make the device nonallocatable by remote users, select No Users.**
- **To make the device allocatable by anyone, select All Users.**

6 If the device is allocatable, *and* your site has created new device authorizations, select the appropriate authorization.

The following dialog box shows the `solaris.device.allocate` authorization is required to allocate the `cdrom0` device.



To create and use site-specific device authorizations, see "Customizing Device Authorizations in Trusted Extensions (Task Map)" on page 337.

7 To save your changes, click OK.

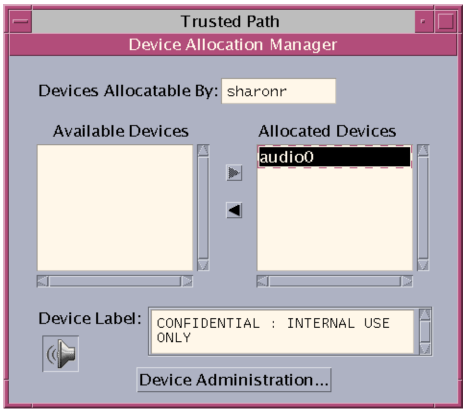
▼ How to Revoke or Reclaim a Device in Trusted Extensions

If a device is not listed in the Device Manager, it might already be allocated or it might be in an allocate error state. The system administrator can recover the device for use.

Before You Begin You must be in the System Administrator role in the global zone. This role includes the `solaris.device.revoke` authorization.

1. **From the Trusted Path menu, select Allocate Device.**

   In the following figure, the audio device is already allocated to a user.



2. **Click the Device Administration button.**

3. **Check the status of a device.**

   Select the device name and check the State field.

   - **If the State field is Allocate Error State, click the Reclaim button.**
   - **If the State field is Allocated, do one of the following:**
     - **Ask the user in the Owner field to deallocate the device.**
     - **Force deallocation of the device by clicking the Revoke button.**

4. **Close the Device Manager.**

▼ How to Protect Nonallocatable Devices in Trusted Extensions

The No Users option in the Allocatable By section of the Device Configuration dialog box is used most often for the frame buffer and printer, which do not have to be allocated to be used.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **From the Trusted Path menu, select Allocate Device.**
2. **In the Device Manager, click the Device Administration button.**
3. **Select the new printer or frame buffer.**
   a. **To make the device nonallocatable, click No Users.**
   b. **(Optional) Restrict the label range on the device.**
      i. **Set the minimum label.**
      Click the Min Label... button. Choose a minimum label from the label builder. For information about the label builder, see "Label Builder in Trusted Extensions" on page 145.
      ii. **Set the maximum label.**
      Click the Max Label... button. Choose a maximum label from the label builder.

Example 23–1 Preventing Remote Allocation of the Audio Device

The No Users option in the Allocatable By section prevents remote users from hearing conversations around a remote system.

The security administrator configures the audio device in the Device Manager as follows:

```
Device Name: audio
For Allocations From: Trusted Path
Allocatable By: Authorized Users
Authorizations: solaris.device.allocate

Device Name: audio
For Allocations From: Non-Trusted Pathh
Allocatable By: No Users
```

▼ How to Configure a Serial Line for Logins

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Open the Solaris Management Console in the Files scope.**

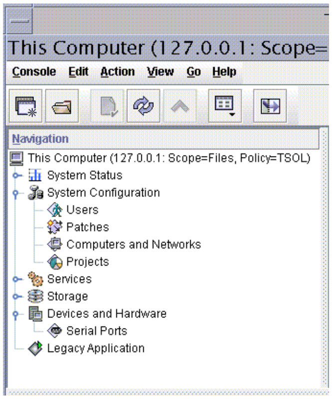


FIGURE 23–1 Serial Ports Tool in the Solaris Management Console

2 **Under Devices and Hardware, navigate to Serial Ports.**

Provide a password when prompted. Follow the online help to configure the serial port.

3 **To change the default label range, open the Device Manager.**

The default label range is `ADMIN_LOW` to `ADMIN_HIGH`.

Example 23–2 Restricting the Label Range of a Serial Port

After creating a serial login device, the security administrator restricts the label range of the serial port to a single label, `Public`. The administrator sets the following values in the Device Administration dialog boxes.

```
Device Name: /dev/term/[a|b]
Device Type: tty
Clean Program: /bin/true
Device Map: /dev/term/[a|b]
Minimum Label: Public
Maximum Label: Public
Allocatable By: No Users
```

▼ How to Add a Device_Clean Script in Trusted Extensions

If no `device_clean` script is specified at the time a device is created, the default script, `/bin/true`, is used.

Before You Begin Have ready a script that purges all usable data from the physical device and that returns `0` for success. For devices with removable media, the script attempts to eject the media if the user does not do so. The script puts the device into the allocate error state if the medium is not ejected. For details about the requirements, see the `device_clean(5)` man page.

You must be in the System Administrator role in the global zone.

1. **Copy the script into the `/etc/security/lib` directory.**

2. **In the Device Administration dialog box, specify the full path to the script.**

   a. **Open the Device Manager.**

   b. **Click the Administration button.**

   c. **Select the name of the device, and click the Configure button.**

   d. **In the Clean Program field, type the full path to the script.**

3. **Save your changes.**

Customizing Device Authorizations in Trusted Extensions (Task Map)

The following task map describes procedures to change device authorizations at your site.

Task	Description	For Instructions
Create new device authorizations.	Creates site-specific authorizations.	"How to Create New Device Authorizations" on page 337
Add authorizations to a device.	Adds site-specific authorizations to selected devices.	"How to Add Site-Specific Authorizations to a Device in Trusted Extensions" on page 341
Assign device authorizations to users and roles.	Enables users and roles to use the new authorizations.	"How to Assign Device Authorizations" on page 341

▼ How to Create New Device Authorizations

If no authorization is specified at the time a device is created, by default, all users can use the device. If an authorization is specified, then, by default, only authorized users can use the device.

To prevent all access to an allocatable device without using authorizations, see Example 23–1.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Edit the `auth_attr` file.**

   Use the trusted editor. For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.

2. **Create a heading for the new authorizations.**

   Use the reverse-order Internet domain name of your organization followed by optional additional arbitrary components, such as the name of your company. Separate components by dots. End heading names with a dot.

   domain-suffix.*domain-prefix*.*optional*.:::*Company* `Header::help=`*Company*`.html`

3. **Add new authorization entries.**

   Add the authorizations, one authorization per line. The lines are split for display purposes. The authorizations include `grant` authorizations that enable administrators to assign the new authorizations.

   domain-suffix.*domain-prefix*`.grant:::Grant All` *Company* `Authorizations::`
   `help=`*Company*`Grant.html`
   domain-suffix.*domain-prefix*`.grant.device:::Grant` *Company* `Device Authorizations::`
   `help=`*Company*`GrantDevice.html`
   domain-suffix.*domain-prefix*`.device.allocate.tape:::Allocate Tape Device::`
   `help=`*Company*`TapeAllocate.html`
   domain-suffix.*domain-prefix*`.device.allocate.floppy:::Allocate Floppy Device::`
   `help=`*Company*`FloppyAllocate.html`

4. **Save the file and close the editor.**

5. **If you are using LDAP as your naming service, update the `auth_attr` entries on the Sun Java System Directory Server (LDAP server).**

   For information, see the `ldapaddent`(1M) man page.

6. **Add the new authorizations to the appropriate rights profiles. Then assign the profiles to users and roles.**

   Use the Solaris Management Console. Assume the Security Administrator role, then follow the Solaris procedure "How to Create or Change a Rights Profile" in *System Administration Guide: Security Services*.

7. **Use the authorization to restrict access to tape and diskette drives.**

   Add the new authorizations to the list of required authorizations in the Device Manager. For the procedure, see "How to Add Site-Specific Authorizations to a Device in Trusted Extensions" on page 341.

Example 23–3 Creating Fine-Grained Device Authorizations

A security administrator for NewCo needs to construct fine-grained device authorizations for the company.

First, the administrator writes the following help files, and places the files in the `/usr/lib/help/auths/locale/C` directory:

```
Newco.html
NewcoGrant.html
NewcoGrantDevice.html
NewcoTapeAllocate.html
NewcoFloppyAllocate.html
```

Next, the administrator adds a header for all of the authorizations for `newco.com` in the `auth_attr` file.

```
# auth_attr file
com.newco.:::NewCo Header::help=Newco.html
```

Next, the administrator adds authorization entries to the file:

```
com.newco.grant:::Grant All NewCo Authorizations::
help=NewcoGrant.html
com.newco.grant.device:::Grant NewCo Device Authorizations::
help=NewcoGrantDevice.html
com.newco.device.allocate.tape:::Allocate Tape Device::
help=NewcoTapeAllocate.html
com.newco.device.allocate.floppy:::Allocate Floppy Device::
help=NewcoFloppyAllocate.html
```

The lines are split for display purposes.

The `auth_attr` entries create the following authorizations:

- An authorization to grant all NewCo's authorizations
- An authorization to grant NewCo's device authorizations
- An authorization to allocate a tape drive
- An authorization to allocate a diskette drive

Example 23–4 Creating Trusted Path and Non-Trusted Path Authorizations

By default, the Allocate Devices authorization enables allocation from the trusted path and from outside the trusted path.

In the following example, site security policy requires restricting remote CD-ROM allocation. The security administrator creates the `com.someco.device.cdrom.local` authorization. This authorization is for CD-ROM drives that are allocated with the trusted path. The `com.someco.device.cdrom.remote` authorization is for those few users who are allowed to allocate a CD-ROM drive outside the trusted path.

The security administrator creates the help files, adds the authorizations to the `auth_attr` database, adds the authorizations to the devices, and then places the authorizations in rights profiles. The profiles are assigned to users who are allowed to allocate devices.

- The following are the `auth_attr` database entries:

  ```
  com.someco.:::SomeCo Header::help=Someco.html
  com.someco.grant:::Grant All SomeCo Authorizations::
  help=SomecoGrant.html
  com.someco.grant.device:::Grant SomeCo Device Authorizations::
  help=SomecoGrantDevice.html
  com.someco.device.cdrom.local:::Allocate Local CD-ROM Device::
  help=SomecoCDAllocateLocal.html
  com.someco.device.cdrom.remote:::Allocate Remote CD-ROM Device::
  help=SomecoCDAllocateRemote.html
  ```

- The following is the Device Manager assignment:

  The Trusted Path enables authorized users to use the Device Manager when allocating the local CD-ROM drive.

  ```
  Device Name: cdrom_0
  For Allocations From: Trusted Path
  Allocatable By: Authorized Users
  Authorizations: com.someco.device.cdrom.local
  ```

  The Non-Trusted Path enables users to allocate a device remotely by using the `allocate` command.

  ```
  Device Name: cdrom_0
  For Allocations From: Non-Trusted Path
  Allocatable By: Authorized Users
  Authorizations: com.someco.device.cdrom.remote
  ```

- The following are the rights profile entries:

  ```
  # Local Allocator profile
  com.someco.device.cdrom.local

  # Remote Allocator profile
  com.someco.device.cdrom.remote
  ```

- The following are the rights profiles for authorized users:

  ```
  # List of profiles for regular authorized user
  Local Allocator Profile
  ...

  # List of profiles for role or authorized user
  ```

```
Remote Allocator Profile
...
```

▼ How to Add Site-Specific Authorizations to a Device in Trusted Extensions

Before You Begin You must be in the Security Administrator role, or in a role that includes the Configure Device Attributes authorization. You must have already created site-specific authorizations, as described in "How to Create New Device Authorizations" on page 337.

1. **Follow the "How to Configure a Device in Trusted Extensions" on page 331 procedure.**

   a. **Select a device that needs to be protected with your new authorizations.**

   b. **Open the Device Administration dialog box.**

   c. **In the Device Configuration dialog box, click the Authorizations button.**

      The new authorizations are displayed in the Not Required list.

   d. **Add the new authorizations to the Required list of authorizations.**

2. **To save your changes, click OK.**

▼ How to Assign Device Authorizations

The Allocate Device authorization enables users to allocate a device. The Allocate Device authorization, and the Revoke or Reclaim Device authorization, are appropriate for administrative roles.

Before You Begin You must be in the Security Administrator role in the global zone.

If the existing profiles are not appropriate, the security administrator can create a new profile. For an example, see "How to Create a Rights Profile for Convenient Authorizations" on page 190.

- **Assign to the user a rights profile that contains the Allocate Device authorization.**

  For assistance, see the online help. For the step-by-step procedure, see "How to Change the RBAC Properties of a User" in *System Administration Guide: Security Services*.

The following profiles enable a role to allocate devices:

- All Authorizations
- Device Management
- Media Backup
- Media Restore
- Object Label Management
- Software Installation

The following profiles enable a role to revoke or reclaim devices:

- All Authorizations
- Device Management

The following profiles enable a role to create or configure devices:

- All Authorizations
- Device Security

Example 23–5 Assigning New Device Authorizations

In this example, the security administrator configures the new device authorizations for the system and assigns the rights profile with the new authorizations to trustworthy users. The security administrator does the following:

1. Creates new device authorizations, as in "How to Create New Device Authorizations" on page 337
2. In the Device Manager, adds the new device authorizations to the tape and diskette drives
3. Places the new authorizations in the rights profile, NewCo Allocation
4. Adds the NewCo Allocation rights profile to the profiles of users and roles who are authorized to allocate tape and diskette drives

Authorized users and roles can now use the tape drives and diskette drives on this system.

CHAPTER 24

Trusted Extensions Auditing (Overview)

This chapter describes the additions to auditing that Solaris Trusted Extensions provides.

Trusted Extensions and Auditing

On a system that is configured with Trusted Extensions software, auditing is configured and is administered similarly to auditing on a Solaris system. However, the following are some differences.

- Trusted Extensions software adds audit classes, audit events, audit tokens, and audit policy options to the system.
- By default, auditing is enabled in Trusted Extensions software.
- Solaris per-zone auditing is not supported. In Trusted Extensions, all zones are audited identically.
- Trusted Extensions provides administrative tools to administer the users' audit characteristics and to edit audit files.
- Two roles, System Administrator and Security Administrator, are used to configure and administer auditing in Trusted Extensions.

  The security administrator plans what to audit and any site-specific, event-to-class mappings. As in the Solaris OS, the system administrator plans disk space requirements for the audit files, creates an audit administration server, and installs audit configuration files.

Audit Management by Role in Trusted Extensions

Auditing in Trusted Extensions requires the same planning as in the Solaris OS. For details about planning, see Chapter 29, "Planning for Solaris Auditing," in *System Administration Guide: Security Services*.

Role Setup for Audit Administration

In Trusted Extensions, auditing is the responsibility of two roles. The System Administrator role sets up the disks and the network of audit storage. The Security Administrator role decides what is to be audited, and specifies the information in the audit configuration files. As in the Solaris OS, you create the roles in software. The rights profiles for these two roles are provided. The initial setup team created the Security Administrator role during initial configuration. For details, see "Create the Security Administrator Role in Trusted Extensions" on page 83.

Note – A system only records the security-relevant events that the audit configuration files configure the system to record (that is, by preselection). Therefore, any subsequent audit review can only consider the events that have been recorded. As a result of misconfiguration, attempts to breach the security of the system can go undetected, or the administrator is unable to detect the user who is responsible for an attempted breach of security. Administrators must regularly analyze audit trails to check for breaches of security.

Audit Tasks in Trusted Extensions

The procedures to configure and manage auditing in Trusted Extensions differ slightly from Solaris procedures:

- Audit configuration is performed in the global zone by one of two administrative roles. Then, the system administrator copies specific customized audit files from the global zone to every labeled zone. By following this procedure, user actions are audited identically in the global zone and in labeled zones.

  For details, see "Audit Tasks of the Security Administrator" on page 345 and "Audit Tasks of the System Administrator" on page 345
- Trusted Extensions administrators use a trusted editor to edit audit configuration files.
- Trusted Extensions administrators use the Solaris Management Console to configure specific users. User-specific audit characteristics can be specified in this tool. Specifying user characteristics is only required when the user's audit characteristics differ from the audit characteristics of the systems on which the user works. For an introduction to the tool, see "Solaris Management Console Tools" on page 141.

Audit Tasks of the Security Administrator

The following tasks are security-relevant, and are therefore the responsibility of the security administrator. Follow the Solaris instructions, but use the Trusted Extensions administrative tools.

Task	For Solaris Instructions	Trusted Extensions Instructions
Configure audit files.	"Configuring Audit Files (Task Map)" in *System Administration Guide: Security Services*	Use the trusted editor. For details, see "How to Edit Administrative Files in Trusted Extensions" on page 155.
(Optional) Change default audit policy.	"How to Configure Audit Policy" in *System Administration Guide: Security Services*	Use the trusted editor.
Disable and re-enable auditing.	"How to Disable the Audit Service" in *System Administration Guide: Security Services*	Auditing is enabled by default.
Manage auditing.	"Solaris Auditing (Task Map)" in *System Administration Guide: Security Services*	Use the trusted editor. Ignore per-zone audit tasks.

Audit Tasks of the System Administrator

The following tasks are the responsibility of the system administrator. Follow the Solaris instructions, but use the Trusted Extensions administrative tools.

Task	For Solaris Instructions	Trusted Extensions Instructions
Create audit partitions and an audit administration server, export audit partitions, and mount audit partitions. Create an `audit_warn` alias.	"Configuring and Enabling the Audit Service (Tasks)" in *System Administration Guide: Security Services*	Perform all administration in the global zone. Use the trusted editor.
Copy or loopback mount customized audit files to labeled zones.	"Configuring the Audit Service in Zones (Tasks)" in *System Administration Guide: Security Services*	Copy the files to the first labeled zone, then copy the zone. Or, loopback mount or copy the files to every labeled zone after the zones are created.
(Optional) Distribute audit configuration files.	No instructions	See "How to Copy Files From Portable Media in Trusted Extensions" on page 99
Manage auditing.	"Solaris Auditing (Task Map)" in *System Administration Guide: Security Services*	Ignore per-zone audit tasks.

Task	For Solaris Instructions	Trusted Extensions Instructions
Select audit records by label.	"How to Select Audit Events From the Audit Trail" in *System Administration Guide: Security Services*	To select records by label, use the `auditreduce` command with the `-l` option.

Trusted Extensions Audit Reference

Trusted Extensions software adds audit classes, audit events, audit tokens, and audit policy options to the Solaris OS. Several auditing commands are extended to handle labels. Trusted Extensions audit records include a label, as shown in the following figure.

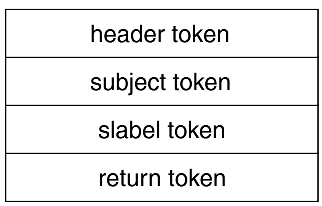


FIGURE 24–1 Typical Audit Record on a Labeled System

Trusted Extensions Audit Classes

The audit classes that Trusted Extensions software adds to the Solaris OS are listed alphabetically in the following table. The classes are listed in the `/etc/security/audit_class` file. For more information about audit classes, see the `audit_class(4)` man page.

TABLE 24–1 X Server Audit Classes

Short Name	Long Name	Audit Mask
xc	X - Object create/destroy	0x00800000
xp	X - Privileged/administrative operations	0x00400000
xs	X - Operations that always silently fail, if bad	0x02000000
xx	X - All X events in the xl, xc, xp, and xs classes (metaclass)	0x03e00000

The X server audit events are mapped to these classes according to the following criteria:

- **xc** – This class audits server objects for creation or for destruction. For example, this class audits `CreateWindow()`.
- **xp** – This class audits for use of privilege. Privilege use can be successful or unsuccessful. For example, `ChangeWindowAttributes()` is audited when a client attempts to change the attributes of another client's window. This class also includes administrative routines such as `SetAccessControl()`.
- **xs** – This class audits routines that do not return X error messages to clients on failure when security attributes cause the failure. For example, `GetImage()` does not return a `BadWindow` error if it cannot read from a window for lack of privilege.

  These events should be selected for audit on success only. When xs events are selected for failure, the audit trail fills with irrelevant records.
- **xx** – This class includes all of the X audit classes.

Trusted Extensions Audit Events

Trusted Extensions software adds audit events to the system. The new audit events and the audit classes to which the events belong are listed in the `/etc/security/audit_event` file. The audit event numbers for Trusted Extensions are between 9000 and 10000. For more information about audit events, see the `audit_event(4)` man page.

Trusted Extensions Audit Tokens

The audit tokens that Trusted Extensions software adds to the Solaris OS are listed alphabetically in the following table. The tokens are also listed in the `audit.log(4)` man page.

TABLE 24–2 Trusted Extensions Audit Tokens

Token Name	Description
"`label` Token" on page 348	Sensitivity label
"`xatom` Token" on page 348	X window atom identification
"`xclient` Token" on page 348	X client identification
"`xcolormap` Token" on page 349	X window color information
"`xcursor` Token" on page 349	X window cursor information
"`xfont` Token" on page 349	X window font information
"`xgc` Token" on page 349	X window graphical context information
"`xpixmap` Token" on page 350	Xwindow pixel mapping information

TABLE 24–2 Trusted Extensions Audit Tokens (Continued)

Token Name	Description
"xproperty Token" on page 350	X window property information
"xselect Token" on page 350	X window data information
"xwindow Token" on page 351	X window window information

`label` Token

The `label` token contains a sensitivity label. This token contains the following fields:

- A token ID
- A sensitivity label

A `label` token is displayed by the `praudit` command as follows:

```
sensitivity label,ADMIN_LOW
```

`xatom` Token

The `xatom` token contains information concerning an X atom. This token contains the following fields:

- A token ID
- The string length
- A text string that identifies the atom

An `xatom` token is displayed by `praudit` as follows:

```
X atom,_DT_SAVE_MODE
```

`xclient` Token

The `xclient` token contains information concerning the X client. This token contains the following fields:

- A token ID
- The client ID

An `xclient` token is displayed by `praudit` as follows:

```
X client,15
```

`xcolormap` Token

The `xcolormap` token contains information about the colormaps. This token contains the following fields:

- A token ID
- The X server identifier
- The creator's user ID

An `xcolormap` token is displayed by `praudit` as follows:

```
X color map,0x08c00005,srv
```

`xcursor` Token

The `xcursor` token contains information about the cursors. This token contains the following fields:

- A token ID
- The X server identifier
- The creator's user ID

An `xcursor` token is displayed by `praudit` as follows:

```
X cursor,0x0f400006,srv
```

`xfont` Token

The `xfont` token contains information about the fonts. This token contains the following fields:

- A token ID
- The X server identifier
- The creator's user ID

An `xfont` token is displayed by `praudit` as follows:

```
X font,0x08c00001,srv
```

`xgc` Token

The `xgc` token contains information about the xgc. This token contains the following fields:

- A token ID
- The X server identifier
- The creator's user ID

An `xgc` token is displayed by `praudit` as follows:

```
Xgraphic context,0x002f2ca0,srv
```

`xpixmap` Token

The `xpixmap` token contains information about the pixel mappings. This token contains the following fields:

- A token ID
- The X server identifier
- The creator's user ID

An `xpixmap` token is displayed by `praudit` as follows:

```
X pixmap,0x08c00005,srv
```

`xproperty` Token

The `xproperty` token contains information about various properties of a window. This token contains the following fields:

- A token ID
- The X server identifier
- The creator's user ID
- A string length
- A text string that identifies the atom

An `xproperty` token is displayed by `praudit` as follows:

```
X property,0x000075d5,root,_MOTIF_DEFAULT_BINDINGS
```

`xselect` Token

The `xselect` token contains the data that is moved between windows. This data is a byte stream with no assumed internal structure and a property string. This token contains the following fields:

- A token ID
- The length of the property string
- The property string
- The length of the property type
- The property type string
- A length field that gives the number of bytes of data
- A byte string that contains the data

An `xselect` token is displayed by `praudit` as follows:

```
X selection,entryfield,halogen
```

xwindow Token

The xwindow token contains information about a window. This token contains the following fields:

- A token ID
- The X server identifier
- The creator's user ID

An xwindow token is displayed by praudit as follows:

```
X window,0x07400001,srv
```

Trusted Extensions Audit Policy Options

Trusted Extensions adds two audit policy options to existing Solaris auditing policy options. List the policies to see the additions:

```
$ auditconfig -lspolicy
...
windata_down Include downgraded window information in audit records

windata_up   Include upgraded window information in audit records
```

Extensions to Auditing Commands in Trusted Extensions

The auditconfig, auditreduce, and auditrecord commands are extended to handle Trusted Extensions information:

- The auditconfig command includes the Trusted Extensions audit policies. For details, see the auditconfig(1M) man page.
- The auditreduce command adds the -l option for filtering records according to the label. For details, see the auditreduce(1M) man page.
- The auditrecord command includes the Trusted Extensions audit events. For details, see the auditrecord(1M) man page.

◆ ◆ ◆ CHAPTER 25

Software Management in Trusted Extensions (Tasks)

This chapter contains information about ensuring that third-party software runs in a trustworthy manner on a system that is configured with Solaris Trusted Extensions.

Adding Software to Trusted Extensions

Any software that can be added to a Solaris system can be added to a system that is configured with Trusted Extensions. Additionally, programs that use Trusted Extensions APIs can be added. Adding software to a Trusted Extensions system is similar to adding software to a Solaris system that is running non-global zones.

For example, packaging issues affect systems that have installed non-global zones. Package parameters define the following:

- **The zone scope of the package –** The scope determines the type of zone in which a specific package can be installed.
- **The visibility of the package –** Visibility determines whether a package must be installed and be identical in all zones.
- **The limitation of the package –** One limitation is whether a package must be installed in the current zone only.

In Trusted Extensions, programs are typically installed in the global zone for use by regular users in labeled zones. For details about installing packages in zones, see Chapter 23, "Packages on an OpenSolaris System With Zones Installed," in *System Administration Guide: Virtualization Using the Solaris Operating System*. Also, see the `pkgadd(1M)` man page.

At a Trusted Extensions site, the system administrator and the security administrator work together to install software. The security administrator evaluates software additions for adherence to security policy. When the software requires privileges or authorizations to succeed, the Security Administrator role assigns an appropriate rights profile to the users of that software.

To import software from removable media requires authorization. An account with the Allocate Device authorization can import or export data from removable media. Data can include executable code. A regular user can only import data at a label within that user's clearance.

The System Administrator role is responsible for adding the programs that the security administrator approves.

Solaris Security Mechanisms for Software

Trusted Extensions uses the same security mechanisms as the Solaris OS. The mechanisms include the following:

- **Authorizations –** Users of a program can be required to have a particular authorization. For information about authorizations, see "Solaris RBAC Elements and Basic Concepts" in *System Administration Guide: Security Services*. Also, see the `auth_attr(4)` and `getauthattr(3SECDB)` man pages.
- **Privileges –** Programs and processes can be assigned privileges. For information about privileges, see Chapter 8, "Using Roles and Privileges (Overview)," in *System Administration Guide: Security Services*. Also, see the `privileges(5)` man page.

  The `ppriv` command provides a debugging utility. For details, see the `ppriv(1)` man page. For instructions on using this utility with programs that work in non-global zones, see "Using the ppriv Utility" in *System Administration Guide: Virtualization Using the Solaris Operating System*.
- **Right Profiles –** Rights profiles collect security attributes in one place for assignment to users or roles. For information about rights profiles, see "RBAC Rights Profiles" in *System Administration Guide: Security Services*.
- **Trusted libraries –** Dynamically shared libraries that are used by `setuid`, `setgid`, and privileged programs can be loaded only from trusted directories. As in the Solaris OS, the `crle` command is used to add a privileged program's shared library directories to the list of trusted directories. For details, see the `crle(1)` man page.

Evaluating Software for Security

When software has been assigned privileges or when it runs with an alternate user ID or group ID, the software becomes *trusted*. Trusted software can bypass aspects of the Trusted Extensions security policy. Be aware that you can make software trusted even though it might not be worthy of trust. The security administrator must wait to give privileges to software until careful scrutiny has revealed that the software uses the privileges in a trustworthy manner.

Programs fall into three categories on a trusted system:

- **Programs that require no security attributes –** Some programs run at a single level and require no privileges. These programs can be installed in a public directory, such as `/usr/local`. For access, assign the programs as commands in the rights profiles of users and roles.
- **Programs that run as root –** Some programs execute with `setuid 0`. Such programs can be assigned an effective UID of `0` in a rights profile. The security administrator then assigns the profile to an administrative role.

Tip – If the application can use privileges in a trustworthy manner, assign the needed privileges to the application, and do not execute the program as `root`.

- **Programs that require privileges –** Some programs might need privileges for reasons that are not obvious. Even if a program is not performing any function that seems to violate system security policy, the program might be doing something internally that violates security. For example, the program could be using a shared log file, or the program could be reading from `/dev/kmem`. For security concerns, see the `mem(7D)` man page.

  Sometimes, an internal policy override is not particularly important to the application's correct operation. Rather, the override provides a convenient feature for users.

  If your organization has access to the source code, check if you can remove the operations that require policy overrides without affecting the application's performance.

Developer Responsibilities When Creating Trusted Programs

Even though a program's developer can manipulate privilege sets in the source code, if the security administrator does not assign the required privileges to the program, the program will fail. The developer and security administrator need to cooperate when creating trusted programs.

A developer who writes a trusted program must do the following:

1. Understand where 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. The program must not violate security policy.
4. Compile the program by using shared libraries that are linked to the program from a trusted directory.

   For additional information, see *Solaris Security for Developers Guide*. For examples of code for Trusted Extensions, see *Solaris Trusted Extensions Developer's Guide*.

Security Administrator Responsibilities for Trusted Programs

The security administrator is responsible for testing and evaluating new software. After determining that the software is trustworthy, the security administrator configures rights profiles and other security-relevant attributes for the program.

The security administrator responsibilities include the following:

1. Make sure that the programmer and the program distribution process is trusted.
2. From one of the following sources, determine which privileges are required by the program:
   - Ask the programmer.
   - Search the source code for any privileges that the program expects to use.
   - Search the source code for any authorizations that the program requires of its users.
   - Use the debugging options to the `ppriv` command to search for use of privilege. For examples, see the `ppriv(1)` man page.
3. Examine the source code to make sure that the code behaves in a trustworthy manner regarding the privileges that the program needs to operate.

   If the program fails to use privilege in a trustworthy manner, and you can modify the program's source code, then modify the code. A security consultant or developer who is knowledgeable about security can modify the code. Modifications might include privilege bracketing or checking for authorizations.

   The assignment of privileges must be manual. A program that fails due to lack of privilege can be assigned privileges. Alternatively, the security administrator might decide to assign an effective UID or GID to make the privilege unnecessary.

Managing Software in Trusted Extensions (Tasks)

Managing software in Trusted Extensions is similar to managing software on a Solaris system that has installed non-global zones. For details about zones, see Part II, "Zones," in *System Administration Guide: Virtualization Using the Solaris Operating System*.

▼ How to Add a Software Package in Trusted Extensions

Before You Begin You must be in a role that can allocate a device.

1 Start from the appropriate workspace.

- **To install a software package in the global zone, stay in the global zone.**

- **To install a software package in a labeled zone, create a workspace at that label.**

  For details, see "How to Change the Label of a Workspace" in *Solaris Trusted Extensions User's Guide*.

2. **Allocate the CD-ROM drive.**

   For details, see "How to Allocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*.

3. **Install the software.**

   For details, see "Where to Find Software Management Tasks" in *System Administration Guide: Basic Administration*.

4. **Deallocate the device when you are finished.**

   For details, see "How to Allocate a Device in Trusted Extensions" in *Solaris Trusted Extensions User's Guide*.

▼ How to Install a Java Archive File in Trusted Extensions

This procedure downloads a Java™ archive (JAR) file to the global zone. From the global zone, the administrator can make it available to regular users.

Before You Begin The security administrator has verified that the source of the Java program is trustworthy, that the method of delivery is secure, and that the program can run in a trustworthy manner.

You are in the System Administrator role in the global zone.

1. **Download the JAR file to the `/tmp` directory.**

   For example, if you are selecting software from `http://www.sunfreeware.com`, use the site's "Solaris `pkg-get` tool" instructions.

2. **Open the File Browser and navigate to the `/tmp` directory.**

3. **Double-click the downloaded file.**

4. **To install the software, answer the questions in the dialog boxes.**

5. **Read the installation log.**

Example 25–1 Downloading a JAR File to a User Label

To limit the security risk, the system administrator downloads the software to a single label within a regular user's accreditation range. Then, the security administrator tests the JAR file at that label. When the software passes the test, the security administrator then downgrades the label to `ADMIN_LOW`. The system administrator installs the software on an NFS server to make it available to all users.

1. First, the system administrator creates a workspace at a user label.
2. In that workspace, he downloads the JAR file.
3. At that label, the security administrator tests the file.
4. Then, the security administrator changes the label of the file to `ADMIN_LOW`.
5. Finally, the system administrator copies the file to an NFS server whose label is `ADMIN_LOW`.

APPENDIX A

Site Security Policy

This appendix discusses site security policy issues, and suggests reference books and web sites for further information:

Creating and Managing a Security Policy

Each Solaris Trusted Extensions site is unique and must determine its own security policy. Perform the following tasks when creating and managing a security policy.

- Establish a security team. The security team needs to have representation from top-level management, personnel management, computer system management and administrators, and facilities management. The team must review Trusted Extensions administrators' policies and procedures, and recommend general security policies that apply to all system users.
- Educate management and administration personnel about the site security policy. All personnel involved in the management and administration of the site must be educated about the security policy. Security policies must not be made available to regular users because this policy information has direct bearing on the security of the computer systems.
- Educate users about Trusted Extensions software and the security policy. All users must be familiar with the *Solaris Trusted Extensions User's Guide*. Because the users are usually the first to know when a system is not functioning normally, the user must become acquainted with the system and report any problems to a system administrator. A secure environment needs the users to notify the system administrators immediately if they notice any of the following:

    - A discrepancy in the last login time that is reported at the beginning of each session
    - An unusual change to file data
    - A lost or stolen human-readable printout
    - The inability to operate a user function
- Enforce the security policy. If the security policy is not followed and enforced, the data contained in the system that is configured with Trusted Extensions is not secure. Procedures must be established to record any problems and the measures that were taken to resolve the incidents.
- Periodically review the security policy. The security team must perform a periodic review of the security policy and all incidents that occurred since the last review. Adjustments to the policy can then lead to increased security.

Site Security Policy and Trusted Extensions

The security administrator must design the Trusted Extensions network based on the site's security policy. The security policy dictates configuration decisions, such as the following:

- How much auditing is done for all users and for which classes of events
- How much auditing is done for users in roles and for which classes of events
- How audit data is managed, archived, and reviewed
- Which labels are used in the system and whether the `ADMIN_LOW` and `ADMIN_HIGH` labels will viewable by regular users
- Which user clearances are assigned to individuals
- Which devices (if any) can be allocated by which regular users
- Which label ranges are defined for systems, printers, and other devices
- Whether Trusted Extensions is used in an evaluated configuration or not

Computer Security Recommendations

Consider the following list of guidelines when you develop a security policy for your site.

- Assign the maximum label of a system that is configured with Trusted Extensions to not be greater than the maximum security level of work being done at the site.
- Manually record system reboots, power failures, and shutdowns in a site log.
- Document file system damage, and analyze all affected files for potential security policy violations.
- Restrict operating manuals and administrator documentation to individuals with a valid need for access to that information.

- Report and document unusual or unexpected behavior of any Trusted Extensions software, and determine the cause.
- If possible, assign at least two individuals to administer systems that are configured with Trusted Extensions. Assign one person the security administrator authorization for security-related decisions. Assign the other person the system administrator authorization for system management tasks.
- Establish a regular backup routine.
- Assign authorizations only to users who need them and who can be trusted to use them properly.
- Assign privileges to programs only they need the privileges to do their work, and only when the programs have been scrutinized and proven to be trustworthy in their use of privilege. Review the privileges on existing Trusted Extensions programs as a guide to setting privileges on new programs.
- Review and analyze audit information regularly. Investigate any irregular events to determine the cause of the event.
- Minimize the number of administration IDs.
- Minimize the number of setuid and setgid programs. Use authorizations, privileges, and roles to execute the program and to prevent misuse.
- Ensure that an administrator regularly verifies that regular users have a valid login shell.
- Ensure that an administrator must regularly verifies that regular users have valid user ID values and not system administration ID values.

Physical Security Recommendations

Consider the following list of guidelines when you develop a security policy for your site.

- Restrict access to the systems that are configured with Trusted Extensions. The most secure locations are generally interior rooms that are not on the ground floor.
- Monitor and document access to systems that are configured with Trusted Extensions.
- Secure computer equipment to large objects such as tables and desks to prevent theft. When equipment is secured to a wooden object, increase the strength of the object by adding metal plates.
- Consider removable storage media for sensitive information. Lock up all removable media when the media are not in use.
- Store system backups and archives in a secure location that is separate from the location of the systems.
- Restrict physical access to the backup and archival media in the same manner as you restrict access to the systems.

- Install a high-temperature alarm in the computer facility to indicate when the temperature is outside the range of the manufacturer's specifications. A suggested range is 10°C to 32°C (50°F to 90°F).
- Install a water alarm in the computer facility to indicate water on the floor, in the subfloor cavity, and in the ceiling.
- Install a smoke alarm to indicate fire, and install a fire-suppression system.
- Install a humidity alarm to indicate too much or too little humidity.
- Consider TEMPEST shielding if machines do not have it. TEMPEST shielding might be appropriate for facility walls, floors, and ceilings.
- Allow only certified technicians to open and close TEMPEST equipment to ensure its ability to shield electromagnetic radiation.
- Check for physical gaps that allow entrance to the facility or to the rooms that contain computer equipment. Look for openings under raised floors, in suspended ceilings, in roof ventilation equipment, and in adjoining walls between original and secondary additions.
- Prohibit eating, drinking, and smoking in computer facilities or near computer equipment. Establish areas where these activities can occur without threat to the computer equipment.
- Protect architectural drawings and diagrams of the computer facility.
- Restrict the use of building diagrams, floor maps, and photographs of the computer facility.

Personnel Security Recommendations

Consider the following list of guidelines when you develop a security policy for your site.

- Inspect packages, documents, and storage media when they arrive and before they leave a secure site.
- Require identification badges on all personnel and visitors at all times.
- Use identification badges that are difficult to copy or counterfeit.
- Establish areas that are prohibited for visitors, and clearly mark the areas.
- Escort visitors at all times.

Common Security Violations

Because no computer is completely secure, a computer facility is only as secure as the people who use it. Most actions that violate security are easily resolved by careful users or additional equipment. However, the following list gives examples of problems that can occur:

- Users give passwords to other individuals who should not have access to the system.
- Users write down passwords, and lose or leave the passwords in insecure locations.

- Users set their passwords to easily guessed words or easily guessed names.
- Users learn passwords by watching other users type a password.
- Unauthorized users remove, replace, or physically tamper with hardware.
- Users leave their systems unattended without locking the screen.
- Users change the permissions on a file to allow other users to read the file.
- Users change the labels on a file to allow other users to read the file.
- Users discard sensitive hardcopy documents without shredding them, or users leave sensitive hardcopy documents in insecure locations.
- Users leave access doors unlocked.
- Users lose their keys.
- Users do not lock up removable storage media.
- Computer screens are visible through exterior windows.
- Network cables are tapped.
- Electronic eavesdropping captures signals emitted from computer equipment.
- Power outages, surges, and spikes destroy data.
- Earthquakes, floods, tornadoes, hurricanes, and lightning destroy data.
- External electromagnetic radiation interference such as sun-spot activity scrambles files.

Additional Security References

Government publications describe in detail the standards, policies, methods, and terminology associated with computer security. Other publications listed here are guides for system administrators of UNIX® systems and are useful in gaining a thorough understanding of UNIX security problems and solutions.

The web also provides resources. In particular, the CERT (`http://www.cert.org`) web site alerts companies and users to security holes in the software. The SANS Institute (`http://www.sans.org/`) offers training, an extensive glossary of terms, and an updated list of top threats from the Internet.

U.S. Government Publications

The U.S. government offers many of its publications on the web. The Computer Security Resource Center (CSRC) of the National Institute of Standards and Technology (NIST) publishes articles on computer security. The following are a sample of the publications that can be downloaded from the NIST site (`http://csrc.nist.gov/index.html`).

- *An Introduction to Computer Security: The NIST Handbook*. SP 800-12, October 1995.
- *Standard Security Label for Information Transfer*. FIPS-188, September 1994.
- Swanson, Marianne and Barbara Guttman. *Generally Accepted Principles and Practices for Securing Information Technology Systems*. SP 800-14, September 1996.
- Tracy, Miles, Wayne Jensen, and Scott Bisker. *Guidelines on Electronic Mail Security*. SP 800-45, September 2002. Section E.7 concerns securely configuring LDAP for mail.
- Wilson, Mark and Joan Hash. *Building an Information Technology Security Awareness and Training Program*. SP 800-61, January 2004. Includes a useful glossary.
- Grace, Tim, Karen Kent, and Brian Kim. *Computer Security Incident Handling Guidelines*. SP 800-50, September 2002. Section E.7 concerns securely configuring LDAP for mail.
- Scarfone, Karen,Wayne Jansen, and Miles Tracy. *Guide to General Server Security* SP 800-123, July 2008.
- Souppaya, Murugiah, John Wack, and Karen Kent. *Security Configuration Checklists Program for IT Products*. SP 800-70, May 2005.

UNIX Security Publications

Chirillo, John and Edgar Danielyan. *Sun™ Certified Security Administration for Solaris™ 9 & 10 Study Guide*. McGraw-Hill/Osborne, 2005.

Garfinkel, Simson, Gene Spafford, and Alan Schwartz. *Practical UNIX and Internet Security, 3rd Edition*. O'Reilly & Associates, Inc, Sebastopol, CA, 2006.

General Computer Security Publications

Brunette, Glenn M. and Christoph L. *Toward Systemically Secure IT Architectures*. Sun Microsystems, Inc, June 2005.

Kaufman, Charlie, Radia Perlman, and Mike Speciner. *Network Security: Private Communication in a Public World, 2nd Edition*. Prentice-Hall, 2002.

Pfleeger, Charles P. and Shari Lawrence Pfleeger. *Security in Computing*. Prentice Hall PTR, 2006.

Privacy for Pragmatists: A Privacy Practitioner's Guide to Sustainable Compliance. Sun Microsystems, Inc, August 2005.

Rhodes-Ousley, Mark, Roberta Bragg, and Keith Strassberg. *Network Security: The Complete Reference*. McGraw-Hill/Osborne, 2004.

Stoll, Cliff. *The Cuckoo's Egg*. Doubleday, 1989.

General UNIX Publications

Bach, Maurice J. *The Design of the UNIX Operating System*. Prentice Hall, Englewood Cliffs, NJ, 1986.

Nemeth, Evi, Garth Snyder, and Scott Seebas. *UNIX System Administration Handbook*. Prentice Hall, Englewood Cliffs, NJ, 1989.

APPENDIX B

Configuration Checklist for Trusted Extensions

This checklist provides an overall view of the major configuration tasks for Solaris Trusted Extensions. The smaller tasks are outlined within the major tasks. The checklist does not replace following the steps in this guide.

Checklist for Configuring Trusted Extensions

The following list summarizes what is required to enable and configure Trusted Extensions at your site. Tasks that are covered elsewhere are cross-referenced.

1. Read.
   - Read the first five chapters of Part II, "Administration of Trusted Extensions."
   - Understand site security requirements.
   - Read "Site Security Policy and Trusted Extensions" on page 360.
2. Prepare.
   - Decide the `root` password.
   - Decide the PROM or BIOS security level.
   - Decide the PROM or BIOS password.
   - Decide if attached peripherals are permitted.
   - Decide if access to remote printers is permitted.
   - Decide if access to unlabeled networks is permitted.
   - Decide the zone creation method.
3. Enable Trusted Extensions.
   a. Install the Solaris OS.
      - For remote administration, install the Developer Group or larger group of Solaris packages.
      - For the Clone Zone creation method, select Custom Install, then lay out a `/zone` partition.

   b. Enable `svc:/system/labeld`, the Trusted Extensions service.

4. If using IPv6, enable IPv6 for Trusted Extensions.
5. If using a DOI different from `1`, set the DOI in the `/etc/system` and the `/etc/security/tsol/tnrhtp` files.
6. (Optional) Create ZFS pool for cloning zones.
7. Configure labels.
   a. Finalize your site's `label_encodings` file.
   b. Check and install the file.
   c. Reboot.
8. Configure interfaces for the global zone and for labeled zones.
9. Configure the Solaris Management Console.
10. Configure the naming service.
    - Use the files naming service, which requires no configuration.
    - Or, configure LDAP
      a. Create either a Trusted Extensions proxy server or a Trusted Extensions LDAP server.
      b. Enable the Solaris Management Console server to accept network connections.
      c. Register the Solaris Management Console with LDAP.
      d. Create an LDAP toolbox for the Solaris Management Console.
11. Configure network connections for LDAP.
    - Assign an LDAP server or proxy server to the `cipso` host type in a remote host template.
    - Assign the local system to the `cipso` host type in a remote host template.
    - Make the local system a client of the LDAP server.
12. Create labeled zones by using the txzonemgr script.
13. Configure the network. See "Configuring Trusted Network Databases (Task Map)" on page 270.
    - Identify single-label hosts and limited-range hosts.
    - Determine the labels to apply to incoming data from unlabeled hosts.
    - Customize the remote host templates.
    - Assign individual hosts to templates.
    - Assign subnets to templates.
14. Establish static routing. See "Configuring Routes and Checking Network Information in Trusted Extensions (Task Map)" on page 283.
15. Configure local users and local administrative roles.
    - To enforce separation of duty, create customized rights profiles.
    - Create the Security Administrator role.
    - Create a local user who can assume the Security Administrator role.
    - Create other roles, and possibly other local users to assume these roles.

16. Create home directories on the NFS server.
    - Create home directories for each user at every label that the user can access.
    - (Optional) Prevent users from reading their lower-level home directories.
17. Configure printing. See "Managing Printing in Trusted Extensions (Task Map)" on page 310.
18. Configure devices. See "Handling Devices in Trusted Extensions (Task Map)" on page 329.
    a. Assign the Device Management profile or the System Administrator profile to a role.
    b. To make devices usable, do one of the following:
       - Per system, make devices allocatable.
       - Assign the Allocate Device authorization to selected users and roles.
19. Configure Solaris features.
    - Configure auditing.
    - Configure security settings.
    - Enable particular LDAP clients to be LDAP administration systems.
    - Configure users in LDAP.
    - Configure network roles in LDAP.
    - Mount and share file systems. See Chapter 17, "Managing and Mounting Files in Trusted Extensions (Tasks)"

APPENDIX C

Quick Reference to Trusted Extensions Administration

Solaris Trusted Extensions interfaces extend the Solaris OS. This appendix provides a quick reference of the differences. For a detailed list of interfaces, including library routines and system calls, see Appendix D, "List of Trusted Extensions Man Pages."

Administrative Interfaces in Trusted Extensions

Trusted Extensions provides interfaces for its software. The following interfaces are available only when Trusted Extensions software is running:

`txzonemgr` script — Provides a menu-based wizard for creating, installing, initializing, and booting labeled zones. The title of the menu is Labeled Zone Manager. This script also provides menu items for networking options, name services options, and for clienting the global zone to an existing LDAP server.

Admin Editor — This trusted editor is used to edit system files. In Trusted GNOME, the editor is invoked from the command line. You provide the file to be edited as the argument, as in:

`/usr/dt/bin/trusted_edit` *filename*

Device Manager — In Trusted Extensions, this GUI is used to administer devices. The Device Administration dialog box is used by administrators to configure devices.

The Device Allocation Manager is used by roles and regular users to allocate devices. The GUI is available from the Trusted Path menu.

Label Builder — This application is invoked when the user can choose a label or a clearance. This application also appears when a role assigns labels or label ranges to devices, zones, users, or roles.

Selection Manager	This application is invoked when an authorized user or authorized role attempts to upgrade or downgrade information.
Trusted Path menu	This menu handles interactions with the trusted computing base (TCB). For example, this menu has a Change Password menu item. In Trusted GNOME, you access the Trusted Path menu by clicking the trusted symbol at the left of the trusted stripe.
Administrative commands	Trusted Extensions provides commands to obtain labels and perform other tasks. For a list of the commands, see "Command Line Tools in Trusted Extensions" on page 146.

Solaris Interfaces Extended by Trusted Extensions

Trusted Extensions adds to existing Solaris configuration files, commands, and GUIs.

Administrative commands	Trusted Extensions adds options to selected Solaris commands. For a list, see Table 8–3.
Configuration files	Trusted Extensions adds two privileges, `net_mac_aware` and `net_mlp`. For the use of `net_mac_aware`, see "Access to NFS Mounted Directories in Trusted Extensions" on page 234. Trusted Extensions adds authorizations to the `auth_attr` database. Trusted Extensions adds executables to the `exec_attr` database. Trusted Extensions modifies existing rights profiles in the `prof_attr` database. It also adds profiles to the database. Trusted Extensions adds fields to the `policy.conf` database. For the fields, see "`policy.conf` File Defaults in Trusted Extensions" on page 176. Trusted Extensions adds audit tokens, audit events, audit classes, and audit policy options. For a list, see "Trusted Extensions Audit Reference" on page 346. Trusted Extensions adds global and Phase 1 transform keywords to the `/etc/inet/ike/config` file for configuring label extensions in IPsec key negotiation. For more information, see "Administration of Labeled IPsec" on page 264 and the `ike.config(4)` man page.

Solaris Management Console	Trusted Extensions adds a Security Templates tool to the Computers and Networks tool set. Trusted Extensions adds a Trusted Network Zones tool to the Computers and Networks tool set. Trusted Extensions adds a Trusted Extensions Attributes tab to the Users tool and the Administrative Roles tool.
Shared directories from zones	Trusted Extensions enables you to share directories from labeled zones. The directories are shared at the label of the zone by creating an `/etc/dfs/dfstab` file from the global zone.

Tighter Security Defaults in Trusted Extensions

Trusted Extensions establishes tighter security defaults than the Solaris OS:

Auditing	By default, auditing is enabled. An administrator can turn off auditing. However, auditing is typically required at sites that install Trusted Extensions.
Devices	By default, device allocation is enabled. By default, device allocation requires authorization. Therefore, by default, regular users cannot use removable media. An administrator can remove the authorization requirement. However, device allocation is typically required at sites that install Trusted Extensions.
Printing	Regular users can print only to printers that include the user's label in the printer's label range. By default, printed output has trailer and banner pages. These pages, and the body pages, include the label of the print job. By default, users cannot print PostScript files.
Roles	Roles are available in the Solaris OS, but their use is optional. In Trusted Extensions, roles are required for proper administration. Making the `root` user a role is possible in the Solaris OS. In Trusted Extensions, the `root` user is made a role to better audit who is acting as superuser.

Limited Options in Trusted Extensions

Trusted Extensions narrows the range of Solaris configuration options:

Desktop	Trusted Extensions offers the Solaris Trusted Extensions (GNOME) desktop.
Naming service	The LDAP naming service is supported. All zones must be administered from one naming service.
Zones	The global zone is an administrative zone. Only the `root` user or a role can enter the global zone. Therefore, administrative interfaces that are available to regular Solaris users are not available to regular Trusted Extensions users. For example, in Trusted Extensions, users cannot bring up the Solaris Management Console. Non-global zones are labeled zones. Users work in labeled zones. All zones must be administered from one naming service.

APPENDIX D

List of Trusted Extensions Man Pages

Solaris Trusted Extensions is a configuration of the Solaris OS. This appendix provides a short description of the Solaris man pages that include Trusted Extensions information.

Trusted Extensions Man Pages in Alphabetical Order

The following man pages describe Trusted Extensions software on a Solaris system. These man pages are relevant only on a system that is configured with Trusted Extensions.

Solaris Man Page	Synopsis
add_allocatable(1M)	Adds entries to allocation databases
atohexlabel(1M)	Converts a human-readable label to its internal text equivalent
blcompare(3TSOL)	Compares binary labels
blminmax(3TSOL)	Determines the bound of two labels
chk_encodings(1M)	Checks the label encodings file syntax
fgetlabel(2)	Gets the file's label
getdevicerange(3TSOL)	Gets the label range of a device
getlabel(1)	Displays the label of files
getlabel(2)	Gets the label of a file
getpathbylabel(3TSOL)	Gets the zone pathname
getplabel(3TSOL)	Gets the label of a process
getuserrange(3TSOL)	Gets the label range of a user
getzoneidbylabel(3TSOL)	Gets zone ID from zone label

getzonelabelbyid(3TSOL)	Gets zone label from zone ID
getzonelabelbyname(3TSOL)	Gets zone label from zone name
getzonepath(1)	Displays the root path of the zone that corresponds to the specified label
getzonerootbyid(3TSOL)	Gets zone root pathname from zone root ID
getzonerootbylabel(3TSOL)	Gets zone root pathname from zone label
getzonerootbyname(3TSOL)	Gets zone root pathname from zone name
hextoalabel(1M)	Converts an internal text label to its human-readable equivalent
labelclipping(3TSOL)	Translates a binary label and clips the label to the specified width
label_encodings(4)	Describes the label encodings file
label_to_str(3TSOL)	Converts labels to human-readable strings
labels(5)	Describes Solaris Trusted Extensions label attributes
libtsnet(3LIB)	Is the Solaris Trusted Extensions network library
libtsol(3LIB)	Is the Solaris Trusted Extensions library
m_label(3TSOL)	Allocates and frees resources for a new label
pam_tsol_account(5)	Checks account limitations that are due to labels
plabel(1)	Gets the label of a process
remove_allocatable(1M)	Removes entries from allocation databases
sel_config(4)	Is the selection rules for copy, cut, paste, and drag-and-drop operations
setflabel(3TSOL)	Moves a file to a zone with the corresponding sensitivity label
smtnrhdb(1M)	Manages entries in the Trusted Extensions networking database
smtnrhtp(1M)	Manages entries in the template database for Trusted Extensions networking
smtnzonecfg(1M)	Manages entries in the configuration database for Trusted Extensions networking in non-global zones

str_to_label(3TSOL)	Parses human-readable strings to a label
tnctl(1M)	Configures Trusted Extensions network parameters
tnd(1M)	Is the trusted network daemon
tninfo(1M)	Displays kernel-level Trusted Extensions network information and statistics
trusted_extensions(5)	Introduces Trusted Extensions
TrustedExtensionsPolicy(4)	Is the configuration file for Trusted Extensions X Server Extension
tsol_getrhtype(3TSOL)	Gets the host type from Trusted Extensions network information
updatehome(1M)	Updates the home directory copy and link files for the current label
XTSOLgetClientAttributes(3XTSOL)	Gets the label attributes of an X client
XTSOLgetPropAttributes(3XTSOL)	Gets the label attributes of a window property
XTSOLgetPropLabel(3XTSOL)	Gets the label of a window property
XTSOLgetPropUID(3XTSOL)	Gets the UID of a window property
XTSOLgetResAttributes(3XTSOL)	Gets all label attributes of a window or a pixmap
XTSOLgetResLabel(3XTSOL)	Gets the label of a window, a pixmap, or a colormap
XTSOLgetResUID(3XTSOL)	Gets the UID of a window or a pixmap
XTSOLgetSSHeight(3XTSOL)	Gets the height of the screen stripe
XTSOLgetWorkstationOwner(3XTSOL)	Gets the ownership of the workstation
XTSOLIsWindowTrusted(3XTSOL)	Determines if a window is created by a trusted client
XTSOLMakeTPWindow(3XTSOL)	Make this window a Trusted Path window
XTSOLsetPolyInstInfo(3XTSOL)	Sets polyinstantiation information
XTSOLsetPropLabel(3XTSOL)	Sets the label of a window property
XTSOLsetPropUID(3XTSOL)	Sets the UID of a window property
XTSOLsetResLabel(3XTSOL)	Sets the label of a window or a pixmap
XTSOLsetResUID(3XTSOL)	Sets the UID of a window, a pixmap, or a colormap

XTSOLsetSessionHI(3XTSOL)	Sets the session high sensitivity label to the window server
XTSOLsetSessionLO(3XTSOL)	Sets the session low sensitivity label to the window server
XTSOLsetSSHeight(3XTSOL)	Sets the height of the screen stripe
XTSOLsetWorkstationOwner(3XTSOL)	Sets the ownership of the workstation

Solaris Man Pages That Are Modified by Trusted Extensions

Solaris Trusted Extensions adds information to the following Solaris man pages.

Solaris Man Page	Trusted Extensions Modification
allocate(1)	Adds options to support allocating a device in a zone and cleaning the device in a windowed environment
auditconfig(1M)	Adds the window policy for labeled information
audit_class(4)	Adds X server audit classes
audit_event(4)	Adds audit events
auditreduce(1M)	Adds a label selector
auth_attr(4)	Adds label authorizations
automount(1M)	Adds the capability to mount, and therefore view, lower-level home directories
cancel(1)	Adds label restrictions to a user's ability to cancel a print job
deallocate(1)	Adds options to support deallocating a device in a zone, cleaning the device in a windowed environment, and specifying the type of device to deallocate
device_clean(5)	Is invoked by default in Trusted Extensions
getpflags(2)	Recognizes the NET_MAC_AWARE and NET_MAC_AWARE_INHERIT process flags
getsockopt(3SOCKET)	Gets the mandatory access control status, SO_MAC_EXEMPT, of the socket
getsockopt(3XNET)	Gets the mandatory access control status, SO_MAC_EXEMPT, of the socket
ifconfig(1M)	Adds the all-zones interface
ikeadm(1M)	Adds a debug flag for labeled IKE processes

`ike.config(4)`	Adds the `label_aware` global parameter and three Phase 1 transform keywords, `single_label`, `multi_label`, and `wire_label`
`in.iked(1M)`	Supports the negotiation of labeled security associations through multilevel UDP ports 500 and 4500 in the global zone
`ipseckey(1M)`	Adds three extensions: `label`, `outer-label`, and `implicit-label`
`is_system_labeled(3C)`	Determines whether the system is configured with Trusted Extensions
`ldaplist(1)`	Adds Trusted Extensions network databases
`list_devices(1)`	Adds attributes, such as labels, that are associated with a device
`lp(1)`	Adds the `-nolabels` option
`lpadmin(1M)`	Adds label restrictions to the administrator's ability to administer printing
`lpmove(1M)`	Adds label restrictions to the administrator's ability to move a print job
`lpq(1B)`	Adds label restrictions to the display of print queue information
`lprm(1B)`	Adds label restrictions to the caller's ability to remove print requests
`lpsched(1M)`	Adds label restrictions to the administrator's ability to stop and restart the print service
`lpstat(1)`	Adds label restrictions to the display of the print service status
`netstat(1M)`	Adds the `-R` option to display extended security attributes
`pf_key(7P)`	Adds labels to IPsec security associations (SAs)
`privileges(5)`	Adds Trusted Extensions privileges, such as `PRIV_FILE_DOWNGRADE_SL`
`prof_attr(4)`	Adds rights profiles, such as Object Label Management
`route(1M)`	Adds the `-secattr` option to add extended security attributes to a route
`setpflags(2)`	Sets the `NET_MAC_AWARE` per-process flag
`setsockopt(3SOCKET)`	Sets the `SO_MAC_EXEMPT` option
`setsockopt(3XNET)`	Sets the mandatory access control, `SO_MAC_EXEMPT`, on the socket
`smrole(1M)`	Adds options to support a role's label

smuser(1M)	Adds options to support a user's label and other security attributes, such as permitted idle time
socket.h(3HEAD)	Supports the SO_MAC_EXEMPT option for unlabeled peers
tar(1)	Adds including labels in tar files and extracting files according to label
tar.h(3HEAD)	Adds attribute types that are used in labeled tar files
ucred_getlabel(3C)	Adds getting the label value on a user credential
user_attr(4)	Adds user security attributes that are specific to Trusted Extensions

Glossary

accreditation range A set of sensitivity labels that are approved for a class of users or resources. A set of valid labels. See also system accreditation range and user accreditation range.

administrative role A role that gives required authorizations, privileged commands, and the Trusted Path security attribute to allow the role to perform administrative tasks. Roles perform a subset of Solaris superuser's capabilities, such as backup or auditing.

allocation A mechanism by which access to a device is controlled. See device allocation.

authorization A right granted to a user or role to perform an action that would otherwise not be allowed by security policy. Authorizations are granted in rights profiles. Certain commands require the user to have certain authorizations to succeed. For example, to print a PostScript file requires the Print Postscript authorization.

CIPSO label Common IP Security Option. CIPSO is the label standard that Trusted Extensions implements.

classification The hierarchical component of a clearance or a label. A classification indicates a hierarchical level of security, for example, `TOP SECRET` or `UNCLASSIFIED`.

clearance The upper limit of the set of labels at which a user can work. The lower limit is the minimum label that is assigned by the security administrator. A clearance can be one of two types, a session clearance or a user clearance.

client A system connected to a network.

closed network A network of systems that are configured with Trusted Extensions. The network is cut off from any non-Trusted Extensions host. The cutoff can be physical, where no wire extends past the Trusted Extensions network. The cutoff can be in the software, where the Trusted Extensions hosts recognize only Trusted Extensions hosts. Data entry from outside the network is restricted to peripherals attached to Trusted Extensions hosts. Contrast with open network.

compartment A nonhierarchical component of a label that is used with the classification component to form a clearance or a label. A compartment represents a collection of information, such as would be used by an engineering department or a multidisciplinary project team.

.copy_files file An optional setup file on a multilabel system. This file contains a list of startup files, such as `.cshrc` or `.mozilla`, that the user environment or user applications require in order for the system or application to behave well. The files that are listed in `.copy_files` are then *copied* to the user's home directory at higher labels, when those directories are created. See also .link_files file.

DAC See discretionary access control.

device Devices include printers, computers, tape drives, floppy drives, CD-ROM drives, DVD drives, audio devices, and internal pseudo terminal devices. Devices are subject to the read equal write equal MAC policy. Access to removable devices, such as DVD drives, are controlled bydevice allocation.

device allocation A mechanism for protecting the information on an allocatable device from access by anybody except the user who allocates the device. Until a device is deallocated, no one but the user who allocated a device can access any information that is associated with the device. For a user to allocate a device, that user must have been granted the Device Allocation authorization by the security administrator.

discretionary access control The type of access that is granted or that is denied by the owner of a file or directory at the discretion of the owner. Solaris Trusted Extensions provides two kinds of discretionary access controls (DAC), UNIX permission bits and ACLs.

domain A part of the Internet naming hierarchy. It represents a group of systems on a local network that share administrative files.

domain name The identification of a group of systems on a local network. A domain name consists of a sequence of component names separated by periods (for example: `example1.town.state.country.org`). As you read a domain name from left to right, the component names identify more general, and usually remote, areas of administrative authority.

domain of interpretation (DOI) On a Solaris system that is configured with Trusted Extensions, the domain of interpretation is used to differentiate between different `label_encodings` files that might have similar labels defined. The DOI is a set of rules that translates the security attributes on network packets to the representation of those security attributes by the local `label_encodings` file. When systems have the same DOI, they share that set of rules and can translate the labeled network packets.

evaluated configuration One or more Trusted Extensions hosts that are running in a configuration that has been certified as meeting specific criteria by a certification authority. In the United States, those criteria are the TCSEC. The evaluating and certifying body is the NSA. Trusted Extensions software that is configured on the Solaris 10 11/06 release is certified to the Common Criteria v2.3 [August 2005], an ISO standard, to Evaluation Assurance Level (EAL) 4, and against a number of protection profiles.

The Common Criteria v2 (CCv2) and protection profiles make the earlier TCSEC U.S. standard obsolete through level B1+. A mutual recognition agreement for CCv2 has been signed by the United States, the United Kingdom, Canada, Denmark, the Netherlands, Germany, and France.

The Trusted Extensions configuration target provides functionality that is similar to the TCSEC C2 and B1 levels, with some additional functionality.

file system A collection of files and directories that, when set into a logical hierarchy, make up an organized, structured set of information. File systems can be mounted from your local system or a remote system.

GFI Government Furnished Information. In this manual, it refers to a U.S. government-provided label_encodings file. In order to use a GFI with Trusted Extensions software, you must add the Sun-specific `LOCAL DEFINITIONS` section to the end of the GFI. For details, see Chapter 5, "Customizing the LOCAL DEFINITIONS Section (Tasks)," in *Solaris Trusted Extensions Label Administration*.

host name The name by which a system is known to other systems on a network. This name must be unique among all the systems within a given domain. Usually, a domain identifies a single organization. A host name can be any combination of letters, numbers, and minus sign (–), but it cannot begin or end with a minus sign.

initial label The minimum label assigned to a user or role, and the label of the user's initial workspace. The initial label is the lowest label at which the user or role can work.

initial setup team A team of at least two people who together oversee the enabling and configuration of Solaris Trusted Extensions software. One team member is responsible for security decisions, and the other for system administration decisions.

IP address Internet protocol address. A unique number that identifies a networked system so it can communicate by means of Internet protocols. In IPv4, the address consists of four numbers separated by periods. Most often, each part of the IP address is a number between 0 and 225. However, the first number must be less than 224 and the last number cannot be 0.

IP addresses are logically divided into two parts: the network, and the system on the network. The network number is similar to a telephone area code. In relation to the network, the system number is similar to a phone number.

label A security identifier that is assigned to an object. The label is based on the level at which the information in that object should be protected. Depending on how the security administrator has configured the user, a user can see the sensitivity label, or no labels at all. Labels are defined in the label_encodings file.

label configuration A Trusted Extensions installation choice of single-label or multilabel sensitivity labels. In most circumstances, label configuration is identical on all systems at your site.

label_encodings file The file where the complete sensitivity label is defined, as are accreditation ranges, label view, default label visibility, default user clearance, and other aspects of labels.

label range A set of sensitivity labels that are assigned to commands, zones, and allocatable devices. The range is specified by designating a maximum label and a minimum label. For commands, the minimum and maximum labels limit the labels at which the command can be executed. Remote hosts that do not recognize labels are assigned a single sensitivity label, as are any other hosts that the security administrator wants to restrict to a single label. A label range limits the labels at which devices can be allocated and restrict the labels at which information can be stored or processed when using the device.

label relationships On a Solaris system that is configured with Trusted Extensions, a label can dominate another label, be equal to another label, or be disjoint from another label. For example, the label `Top Secret` dominates the label `Secret`. For two systems with the same domain of interpretation (DOI), the label `Top Secret` on one system is equal to the label `Top Secret` on the other system.

label set See security label set.

labeled host A labeled system that is part of a trusted network of labeled systems.

labeled system A labeled system is a system that is running a multilevel operating system, such as Trusted Extensions or SELinux with MLS enabled. The system can send and receive network packets that are labeled with a Common IP Security Option (CIPSO) in the header of the packet.

labeled zone On a Solaris system that is configured with Trusted Extensions, every zone is assigned a unique label. Although the global zone is labeled, *labeled zone* typically refers to a non-global zone that is assigned a label. Labeled zones have two different characteristics from non-global zones on a Solaris system that is not configured with labels. First, labeled zones must use the same pool of user IDs and group IDs. Second, labeled zones can share IP addresses.

.link_files file An optional setup file on a multilabel system. This file contains a list of startup files, such as `.cshrc` or `.mozilla`, that the user environment or user applications require in order for the system or application to behave well. The files that are listed in `.link_files` are then *linked* to the user's home directory at higher labels, when those directories are created. See also .copy_files file.

MAC See mandatory access control.

mandatory access control Access control that is based on comparing the sensitivity label of a file, directory, or device to the sensitivity label of the process that is trying to access it. The MAC rule, read equal–read down, applies when a process at one label attempts to read a file at a lower label. The MAC rule, write equal-read down, applies when a process at one label attempts to write to a directory at another label.

minimum label The lower bound of a user's sensitivity labels and the lower bound of the system's sensitivity labels. The minimum label set by the security administrator when specifying a user's security attributes is the sensitivity label of the user's first workspace at first login. The sensitivity label that is specified in the minimum label field by the security administrator in the `label_encodings` file sets the lower bound for the system.

multilevel desktop On a Solaris system that is configured with Trusted Extensions, users can run a desktop at a particular label. If the user is authorized to work at more than one label, the user can create a separate workspace to work at each label. On this multilevel desktop, authorized users can cut and paste between windows at different labels, receive mail at different labels, and view and use labeled windows in workspaces of a different label.

multilevel port (MLP) On a Solaris system that is configured with Trusted Extensions, an MLP is used to provide multilevel service in a zone. By default, the X server is a multilevel service that is defined in the global zone. An MLP is specified by port number and protocol. For example, the MLP of the X server for the multilevel desktop is specified by 6000-6003 and TCP.

naming service A distributed network database that contains key system information about all the systems on a network, so that the systems can communicate with each other. With a naming service, the system information can be maintained, managed, and accessed on a network-wide basis. Sun supports the LDAP naming service. Without such a service, each system has to maintain its own copy of the system information in the local `/etc` files.

networked systems A group of systems that are connected through hardware and software, sometimes referred to as a local area network (LAN). One or more servers are usually needed when systems are networked.

non-networked systems Computers that are not connected to a network or do not rely on other hosts.

open network A network of Solaris Trusted Extensions hosts that is connected physically to other networks and that uses Trusted Extensions software to communicate with non-Trusted Extensions hosts. Contrast with closed network.

outside the evaluated configuration When software that has been proved to be able satisfy the criteria for an evaluated configuration, is configured with settings that do not satisfy security criteria, the software is described as being *outside the evaluated configuration*.

permission bits A type of discretionary access control in which the owner specifies 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 the owner's group, and one set for all others.

primary administrator The person who is entrusted to create new rights profiles for the organization, and to fix machine difficulties that are beyond the power of the security administrator and system administrator combined. This role should be assumed rarely. After initial security configuration, more secure sites can choose not to create this role, and not to assign any role the Primary Administrator profile.

privilege Powers that are granted to a process that is executing a command. The full set of privileges describes the full capabilities of the system, from basic capabilities to administrative capabilities. Privileges that bypass security policy, such as setting the clock on a system, can be granted by a site's security administrator.

process An action that executes a command on behalf of the user who invokes the command. A 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 that are available to the command being executed and the sensitivity label of the current workspace.

profile shell A special shell that recognizes privileges. A profile shell typically limits users to fewer commands, but can allow these commands to run with privilege. The profile shell is the default shell of a trusted role.

remote host A different system than the local system. A remote host can be an unlabeled host or a labeled host.

rights profile A bundling mechanism for commands and for the security attributes that are assigned to these executables. Rights profiles allow Solaris administrators to control who can execute which commands and to control the attributes these commands have when they are executed. When a user logs in, all rights assigned to that user are in effect, and the user has access to all the commands and authorizations assigned in all of that user's rights profiles.

role A role is like a user, except that a role cannot log in. Typically, a role is used to assign administrative capabilities. Roles are limited to a particular set of commands and authorizations . See administrative role.

security administrator In an organization where sensitive information must be protected, the person or persons who define and enforce the site's security policy. These persons are cleared to access all information that is being processed at the site. In software, the Security Administrator administrative role is assigned to one or more individuals who have the proper clearance. These administrators configure the security attributes of all users and hosts so that the software enforces the site's security policy. In contrast, see system administrator.

security attribute An attribute that is used to enforce Trusted Extensions security policy. Various sets of security attributes are assigned to processes, users, zones, hosts, allocatable devices, and other objects.

security label set Specifies a discrete set of security labels for a tnrhtp database entry. Hosts that are assigned to a template with a security label set can send and receive packets that match any one of the labels in the label set.

security policy On a Trusted Extensions host, the set of DAC, MAC, and labeling rules that define how information can be accessed. At a customer site, 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 security label that is assigned to an object or a process. The label is used to limit access according to the security level of the data that is contained.

separation of duty The security policy that two administrators or roles be required to create and authenticate a user. One administrator or role is responsible for creating the user, the user's home directory, and other basic administration. The other administrator or role is responsible for the user's security attributes, such as the password and the label range.

Solaris Management Console A Java-based administrative GUI that contains toolboxes of administrative programs. Most system, network, and user administration is done by using the Console toolboxes.

system Generic name for a computer. After installation, a system on a network is often referred to as a host.

system accreditation range The set of all valid labels that are created according to the rules that the security administrator defines in the label_encodings file, plus the two administrative labels that are used on every system that is configured with Trusted Extensions. The administrative labels are `ADMIN_LOW` and `ADMIN_HIGH`.

system administrator In Trusted Extensions, the trusted role assigned to the user or users who are responsible for performing standard system management tasks such as setting up the non-security-relevant portions of user accounts. In contrast, see security administrator.

tnrhdb database The trusted network remote host database. This database assigns a set of label characteristics to a remote host. The database is accessible either as a file in `/etc/security/tsol/tnrhdb` or from the LDAP server.

tnrhtp database The trusted network remote host template. This database defines the set of label characteristics that a remote host can be assigned. The database is accessible either as a file in `/etc/security/tsol/tnrhtp`, or from the LDAP server.

toolbox A collection of programs in the Solaris Management Console. On a Trusted Extensions host, administrators use `Policy=TSOL` toolboxes. Each toolbox has programs that are usable in the scope of the toolbox. For example, the Trusted Network Zones tool, which handles the system's `tnzonecfg` database, exists only in the `Files` toolbox, because its scope is always local. The User Accounts program exists in all toolboxes. To create a local user, the administrator uses the `Files` toolbox, and to create a network user, the administrator uses the `LDAP` toolbox.

trusted editor On a Solaris system that is configured with Trusted Extensions, the trusted editor is used to create and modify administrative files. The file name cannot be changed by the editor. Also, use of the editor is audited and shell escape commands are disabled. In Trusted GNOME, the `/usr/dt/bin/trusted_edit` command starts the trusted editor.

Trusted Network databases `tnrhtp`, the trusted network remote host template and `tnrhdb`, the trusted network remote host database together define the remote hosts that a Trusted Extensions system can communicate with.

trusted path On a Solaris system that is configured with Trusted Extensions, the trusted path is a reliable, tamper-proof way to interact with the system. The trusted path is used to ensure that administrative functions cannot be compromised. User functions that must be protected, such as changing a password, also use the trusted path. When the trusted path is active, the desktop displays a tamper-proof indicator.

trusted role See administrative role.

trusted stripe A region that cannot be spoofed. In Trusted GNOME the stripe is at the top. The stripe provides visual feedback about the state of the window system: a trusted path indicator and window sensitivity label. When sensitivity labels are configured to not be viewable for a user, the trusted stripe is reduced to an icon that displays only the trusted path indicator.

txzonemgr script The `/usr/sbin/txzonemgr` script provides a simple GUI for managing labeled zones. The script also provides menu items for networking options, name services options, and for clienting the global zone to an existing LDAP server. `txzonemgr` is run by root in the global zone.

unlabeled host A networked system that sends unlabeled network packets, such as a system that is running the Solaris OS.

unlabeled system To a Solaris system that is configured with Trusted Extensions, an unlabeled system is a system that is not running a multilevel operating system, such as Trusted Extensions or SELinux with MLS enabled. An unlabeled system does not send labeled packets. If the communicating Trusted Extensions system has assigned to the unlabeled system a single label, then network communication between the Trusted Extensions system and the unlabeled system happens at that label. An unlabeled system is also called a "single-level system".

user accreditation range The set of all possible labels at which a regular user can work on the system. The site's security administrator specifies the range in the label_encodings file file. The rules for well-formed labels that define the system accreditation range are additionally restricted by the values in the `ACCREDITATION RANGE` section of the file: the upper bound, the lower bound, the combination constraints and other restrictions.

user clearance The clearance assigned by the security administrator that sets the upper bound of the set of labels at which a user can work at any time. The user can decide to accept the default, or can further restrict that clearance during any particular login session.

Index

M

R

S

V

W

X

Z