Trusted Extensions Configuration and Administration

Part No: 821–1482–11
December 2011

ORACLE

111206@25097

Contents

Figures

Tables

Preface

Trusted Extensions Configuration and Administration provides procedures for enabling and initially configuring the Trusted Extensions feature on the Oracle Solaris operating system (Oracle Solaris OS). This guide also provides procedures for managing users, zones, devices, and hosts on a Trusted Extensions system.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the *Oracle Solaris OS: Hardware Compatibility Lists*. This document cites any implementation differences between the platform types.

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, determine who can perform the configuration tasks.

Administrators should be familiar with Oracle 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 *Trusted Extensions User's Guide*
- How administrative tasks are divided among roles at your site

Trusted Extensions and the Oracle Solaris Operating System

Trusted Extensions runs on top of the Oracle Solaris OS. Because Trusted Extensions software can modify the Oracle Solaris OS, Trusted Extensions can require specific settings for Oracle Solaris installation options. Part I of this guide describes how to prepare the Oracle 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 Trusted Extensions Guides Are Organized

The following table lists the topics that are covered in the Trusted Extensions guides and the audience for each guide.

Title of Guide	Topics	Audience
Trusted Extensions User's Guide	Describes the basic features of Trusted Extensions. This guide contains a glossary.	End users, administrators, developers
Trusted Extensions Configuration and Administration	Part I describes how to prepare for, enable, and initially configure Trusted Extensions. Part II describes how to administer a Trusted Extensions system. This guide contains a glossary.	Administrators, developers
Trusted Extensions Developer's Guide	Describes how to develop applications with Trusted Extensions.	Developers, administrators
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 System Administration Guides

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

Book Title	Topics
Booting and Shutting Down Oracle Solaris on SPARC Platforms	Booting and shutting down a system, managing boot services, modifying boot behavior, booting from ZFS, managing the boot archive, and troubleshooting booting on SPARC platforms
Booting and Shutting Down Oracle Solaris on x86 Platforms	Booting and shutting down a system, managing boot services, modifying boot behavior, booting from ZFS, managing the boot archive, and troubleshooting booting on x86 platforms
Oracle Solaris Administration: Common Tasks	Using Oracle Solaris commands, booting and shutting down a system, managing user accounts and groups, managing services, hardware faults, system information, system resources, and system performance, managing software, printing, the console and terminals, and troubleshooting system and software problems

Book Title	Topics
Oracle Solaris Administration: Devices and File Systems	Removable media, disks and devices, file systems, and backing up and restoring data
Oracle Solaris Administration: IP Services	TCP/IP network administration, IPv4 and IPv6 address administration, DHCP, IPsec, IKE, IP Filter, and IPQoS
Oracle Solaris Administration: Naming and Directory Services	DNS, NIS, and LDAP naming and directory services, including transitioning from NIS to LDAP
Oracle Solaris Administration: Network Interfaces and Network Virtualization	Automatic and manual IP interface configuration including WiFi wireless; administration of bridges, VLANs, aggregations, LLDP, and IPMP; virtual NICs and resource management.
Oracle Solaris Administration: Network Services	Web cache servers, time-related services, network file systems (NFS and autofs), mail, SLP, and PPP
Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management	Resource management features, which enable you to control how applications use available system resources; Oracle Solaris Zones software partitioning technology, which virtualizes operating system services to create an isolated environment for running applications; and Oracle Solaris 10 Zones, which host Oracle Solaris 10 environments running on the Oracle Solaris 11 kernel
Oracle Solaris Administration: Security Services	Auditing, device management, file security, BART, Kerberos services, PAM, Cryptographic Framework, Key Management, privileges, RBAC, SASL, Secure Shell, and virus scanning
Oracle Solaris Administration: SMB and Windows Interoperability	SMB service, which enables you to configure an Oracle Solaris system to make SMB shares available to SMB clients; SMB client, which enables you to access SMB shares; and native identity mapping services, which enables you to map user and group identities between Oracle Solaris systems and Windows systems
Oracle Solaris Administration: ZFS File Systems	ZFS storage pool and file system creation and management, snapshots, clones, backups, using access control lists (ACLs) to protect ZFS files, and using ZFS on an Oracle Solaris system with zones installed
Trusted Extensions Configuration and Administration	System installation, configuration, and administration that is specific to Trusted Extensions
Oracle Solaris 11 Security Guidelines	Securing an Oracle Solaris system, as well as usage scenarios for its security features, such as zones, ZFS, and Trusted Extensions
Transitioning From Oracle Solaris 10 to Oracle Solaris 11	Provides system administration information and examples for transitioning from Oracle Solaris 10 to Oracle Solaris 11 in the areas of installation, device, disk, and file system management, software management, networking, system management, security, virtualization, desktop features, user account management, and user environments emulated volumes, and troubleshooting and data recovery

Related References

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 – Oracle is not responsible for the availability of third-party web sites that are mentioned in this document. Oracle 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. Oracle 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.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface	Description	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:`

TABLE P–1 Typographic Conventions *(Continued)*

Typeface	Description	Example
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 Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle 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

The chapters in this part describe how to prepare Oracle Solaris systems to run Trusted Extensions. The chapters cover enabling Trusted Extensions and the initial configuration tasks.

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 Oracle Solaris systems.

Chapter 2, "Configuration Roadmap for Trusted Extensions," provides task maps for configuring Trusted Extensions software on Oracle Solaris systems.

Chapter 3, "Adding the Trusted Extensions Feature to Oracle Solaris (Tasks)," provides instructions on preparing an Oracle Solaris system for Trusted Extensions software. It describes how to enable Trusted Extensions and log in.

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 the LDAP naming service on Trusted Extensions systems.

◆ ◆ ◆ CHAPTER 1

Security Planning for Trusted Extensions

The Trusted Extensions feature of Oracle Solaris 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 two Oracle Solaris features:

- Capabilities that in most UNIX environments are assigned to superuser are handled by discrete administrative roles.
- The ability to override security policy can be assigned to specific users and applications.

In Trusted Extensions, 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. These labels supply mandatory access control (MAC), in addition to UNIX permissions, or discretionary access control (DAC).

Understanding Your Site's Security Policy

Trusted Extensions effectively enables you to integrate your site's security policy with the Oracle Solaris OS. Thus, you need to have a good understanding of the scope of your policy and how Trusted Extensions software can implement 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` role or the System Administrator role is responsible for enabling 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.
- 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 administrative 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 with Trusted Extensions software. 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 Oracle-specific local extensions, at least the `COLOR NAMES` section.

Caution – If you are supplying a `label_encodings` file, best practice is to have the final version of the file installed before the labels are verified by the system. Labels are verified during the first boot after the Trusted Extensions service is enabled. After you create your first zone or network template, any changes to the `label_encodings` file must accommodate the existing zones and templates.

Planning labels also involves planning the label configuration. After enabling the Trusted Extensions service, you need to decide if the system must allow logins at multiple labels, or if the system can be configured with one user label only. For example, an LDAP server is a good candidate to have one labeled zone. For local administration of the server, you would create a zone at the minimum label. To administer the system, the administrator logs in, and from the user workspace assumes the appropriate role.

For more information, see *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.

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 an Oracle Solaris release, as described in *Installing Oracle Solaris 11 Systems* and the Installation section of the *Release Notes* for this release..
- Trusted Extensions features can add to those recommendations:
  - Memory beyond the suggested minimum is required on the following systems:
    - 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 1, "Planning the Network Deployment," in *Oracle Solaris Administration: IP Services*.

Trusted Extensions software recognizes two host types, cipso 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`	Is used to identify untrusted hosts that can communicate with the global zone. Such hosts send packets that do not include labels. For more information, see unlabeled system.

TABLE 1–1 Default Host Templates in Trusted Extensions *(Continued)*

Host Type	Template Name	Purpose
`cipso`	`cipso`	Is used to identify 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 labeling of hosts, gateways, and networks is explained in Chapter 16, "Managing Networks in Trusted Extensions (Tasks)." Assigning labels to remote systems is performed after initial setup.

Planning for Zones in Trusted Extensions

Trusted Extensions software is added to Oracle Solaris 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 in your `label_encodings` file. A provided script enables you to easily create two labeled zones for the default user label and the default user clearance in your `label_encodings` file.

After labeled zones are created, regular users can use the configured system, but they are not connected to other systems.

- In Trusted Extensions, the local transport to connect to the X server is UNIX domain sockets. By default, the X server does not listen for TCP connections.
- By default, non-global zones cannot communicate with untrusted hosts. You must specify the explicit remote host IP addresses or network masks that can be reached by each zone.

Trusted Extensions Zones and Oracle Solaris Zones

Trusted Extensions zones, that is, labeled zones, are a *brand* of Oracle Solaris Zones. Labeled zones are primarily used to segregate data. In Trusted Extensions, regular users cannot remotely log in to a labeled zone except from an equally labeled zone on another trusted system. Authorized administrators can access a labeled zone from the global zone. For more about zone brands, see the `brands(5)` man page.

Zone Creation in Trusted Extensions

Zone creation in Trusted Extensions is similar to zone creation in Oracle Solaris. Trusted Extensions provides the `txzonemgr` script to step you through the process. The script has several command line options to automate the creation of labeled zones.

Access to Labeled Zones

On a properly configured system, every zone must be able to use a network address to communicate with other zones that share the same label. The following configurations provide labeled zone access to other labeled zones:

- **all-zones interface –** One `all-zones` address is assigned. In this default configuration, only one IP address is required. Every zone, global and labeled, can communicate with identically labeled zones on remote systems over this shared address.

  A refinement of this configuration is to create a second IP instance for the global zone to use exclusively. This second instance would not be an `all-zones` address. The IP instance could be used to host a multilevel service or to provide a route to a private subnet.

- **IP instances –** As in the Oracle Solaris OS, one IP address is assigned to every zone, including the global zone. The zones share the IP stack. In the simplest case, all zones share the same physical interface.

  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.

  A further refinement is to use one or more `all-zones` interfaces in addition to an IP instance per zone. This configuration provides the option of using internal interfaces, such as `vni0`, to reach the global zone, thus protecting the global zone from remote attack. For example, a privileged service that binds a multilevel port on an instance of `vni0` in the global zone can only be reached internally by zones that use the shared stack.

- **Exclusive IP stack –** As in Oracle Solaris, one IP address is assigned to every zone, including the global zone. A virtual network interface card (VNIC) is created for each labeled zone.

  A refinement of this configuration is to create each VNIC over a separate network interface. Such a configuration is used to physically separate the single-label networks that are associated with each NIC. Zones that are configured with an exclusive IP stack cannot use the `all-zones` interface.

Planning for Multilevel Services

By default, Trusted Extensions does not provide multilevel services. Most services are easily configured as zone-to-zone services, that is, as single-label services. For example, each labeled zone can connect to the NFS server that runs at the label of the labeled zone.

If your site requires multilevel services, these services are best configured on a system with at least two IP addresses. The multilevel ports that a multilevel service requires can be assigned to the IP address that is associated with the global zone. An `all-zones` address can be used by the labeled zones to reach the services.

Tip – If users in labeled zones must not have access to multilevel services, then you can assign one IP address to the system. A typical use of this Trusted Extensions 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 are planning to use LDAP, your systems must be configured as LDAP clients before you add the first labeled zone.

If you plan to run Trusted Extensions on a network of systems, use LDAP as the naming service. For Trusted Extensions. a populated Oracle Directory Server Enterprise Edition (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 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 enabled when Trusted Extensions is first booted. Therefore, by default, all events in the `login/logout` class are audited. To audit the users who are configuring the system, you can create roles early in the configuration process. When these roles configure the system, the audit records include the login user who assumes the role. See "Creating Roles and Users in Trusted Extensions" on page 66.

Planning auditing in Trusted Extensions is the same as in the Oracle Solaris OS. For details, see Part VII, "Auditing in Oracle Solaris," in *Oracle Solaris Administration: 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 22, "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 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
	CRYPT_ALGORITHMS_ALLOW	1,2a,md5,5,6
	CRYPT_DEFAULT	sha256
	LOCK_AFTER_RETRIES	no | yes
	PRIV_DEFAULT	basic
	PRIV_LIMIT	all
	AUTHS_GRANTED	solaris.device.cdrw
	CONSOLE_USER	Console User
	PROFS_GRANTED	Basic Solaris User
LOCAL DEFINITIONS section of /etc/security/tsol/label_encodings	Default User Clearance	CNF INTERNAL USE ONLY
	Default User Sensitivity Label	PUBLIC

Note – The IDLECMD and IDLETIME variables apply to the login user's session. If the login user assumes a role, the user's IDLECMD and IDLETIME values are in effect for that role.

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 bash shell. The system administrator can set up a template that gives each user a pfbash shell.

Devising a Configuration Strategy for Trusted Extensions

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 discrete roles, and local users who can assume those 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 users log in and assume a limited role. 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.

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

  Early in the configuration process, the `root` role creates additional roles. The `root` role also sets up auditing to audit events that are executed by roles. Once these additional roles have been assigned to the initial user, and the computer is rebooted, the user logs in and assume the appropriate role for the current task. The audit trail provides a record of the configuration process.

- One person enables and configures the software by assuming the `root` role. The configuration process is not audited.

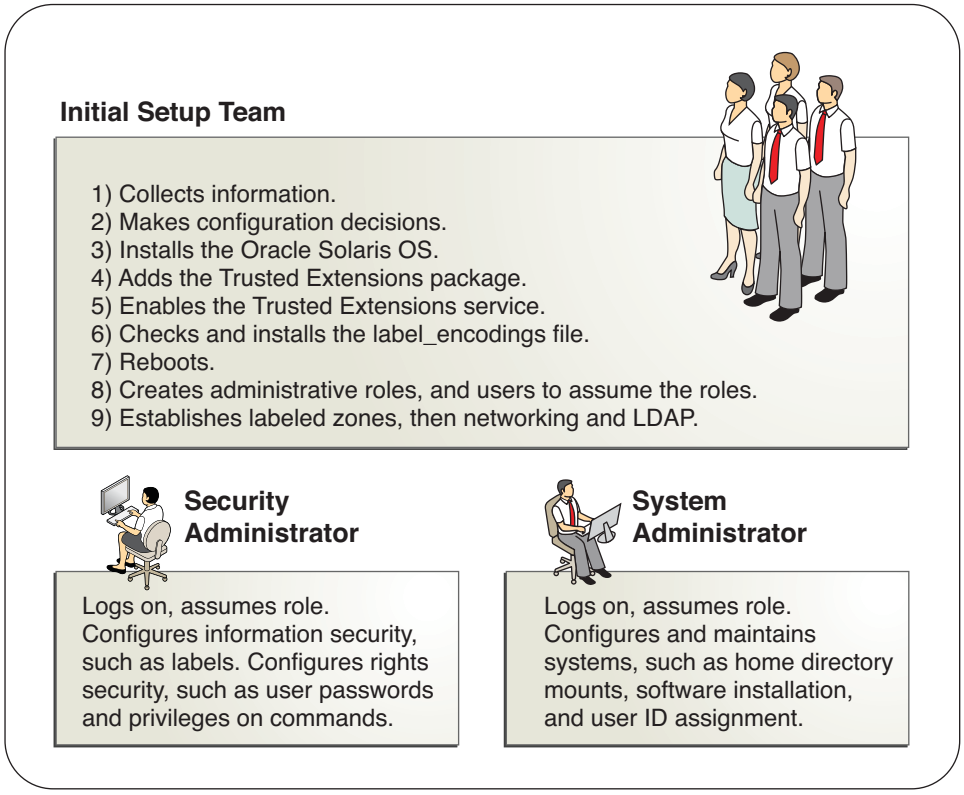
  By using this strategy, no record is kept of the configuration process.

- The initial setup team changes the `root` role into a user.

  No record is kept in the software of the name of the user who is acting as `root`. This setup might be required for remote administration of a headless system.

Task division by role is shown in the following figure. The security administrator configures 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



Resolving Additional Issues Before Enabling Trusted Extensions

Before configuring Trusted Extensions, you must physically protect your systems, decide which labels to attach to zones, and resolve other security issues. For the procedures, see "Resolving Security Issues Before Enabling Trusted Extensions" on page 46.

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 service. 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.

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.

- An Oracle label_encodings file is installed and configured.
- A trusted desktop, Solaris Trusted Extensions (GNOME), creates a labeled windowing environment that provides administrative workspaces in the global zone. These workspaces are protected by the Trusted Path, visible in the trusted stripe.
- As in the Oracle Solaris OS, rights profiles for roles are defined. As in the Oracle Solaris OS, `root` is the only defined role.

  To use additional 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 `tncfg` command enables administrators to view and modify these trusted databases.
- Trusted Extensions provides GUIs to administer the system. For a full list, see Chapter 7, "Trusted Extensions Administration Tools."
  - The `txzonemgr` script enables administrators to configure Trusted Extensions zones and networking. For more information, see the `txzonemgr(1M)` man page.
  - The Device Manager manages the allocation and labeling of attached devices.

◆ ◆ ◆ CHAPTER 2

Configuration Roadmap for Trusted Extensions

This chapter outlines the tasks for enabling and configuring the Trusted Extensions feature of Oracle Solaris.

Caution – If you are enabling and configuring Trusted Extensions remotely, carefully review Chapter 12, "Remote Administration in Trusted Extensions (Tasks)," before booting into the Trusted Extensions environment.

Task Map: Preparing for and Enabling Trusted Extensions

To prepare your system and enable Trusted Extensions, complete the following tasks.

Task	For Instructions
■ Prepare an existing Oracle Solaris installation for Trusted Extensions ■ Install the Oracle Solaris OS with Trusted Extensions in mind.	■ "Prepare an Installed Oracle Solaris System for Trusted Extensions" on page 44 ■ "Install an Oracle Solaris System Securely" on page 44
Gather information and make decisions about your system and your Trusted Extensions network.	"Resolving Security Issues Before Enabling Trusted Extensions" on page 46
Enable Trusted Extensions.	"Enable Trusted Extensions and Reboot" on page 48

Task Map: Choosing a Trusted Extensions Configuration

Configure Trusted Extensions on your system using one of the methods in the following task map.

Task	For Instructions
Create a demonstration Trusted Extensions system.	"Task Map: Configuring Trusted Extensions With the Provided Defaults" on page 40
Create an enterprise Trusted Extensions system.	"Task Map: Configuring Trusted Extensions to Meet Your Site's Requirements" on page 40
Configure Trusted Extensions on a remote system.	Enable Trusted Extensions but do not reboot. Follow instructions in Chapter 12, "Remote Administration in Trusted Extensions (Tasks)." Then, continue with the instructions for systems with a monitor.

Task Map: Configuring Trusted Extensions With the Provided Defaults

For a default configuration, perform the following tasks in sequence.

Task	For Instructions
Load the Trusted Extensions packages.	"Add Trusted Extensions Packages to an Oracle Solaris System" on page 45
Enable Trusted Extensions and reboot.	"Enable Trusted Extensions and Reboot" on page 48
Log in.	"Log In to Trusted Extensions" on page 49
Create two labeled zones.	"How to Create a Default Trusted Extensions System" on page 56 Or, "How to Create Labeled Zones Interactively" on page 57
Create labeled workspaces for the zones.	"How to Assign Labels to Two Zone Workspaces" on page 59

Task Map: Configuring Trusted Extensions to Meet Your Site's Requirements

Tip – For a secure configuration process, create roles early in the process.

The order of tasks is shown in the following task map.

- The tasks in "Creating Labeled Zones" on page 55 are required.
- Depending on your site's requirements, perform other configuration tasks.

Task	For Instructions
Configure the global zone.	"Setting Up the Global Zone in Trusted Extensions" on page 51
Configure the labeled zones.	"Creating Labeled Zones" on page 55
To communicate with other systems, set up networking.	"Configuring the Network Interfaces in Trusted Extensions" on page 60
Configure the LDAP naming service. **Note –** Skip if you are not using LDAP.	Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)"
Complete system configuration.	Part II, "Administration of Trusted Extensions"

◆ ◆ ◆ CHAPTER 3

Adding the Trusted Extensions Feature to Oracle Solaris (Tasks)

This chapter describes how to prepare for and enable the Trusted Extensions service on an Oracle Solaris system. This chapter covers the following topics:

Initial Setup Team Responsibilities

Trusted Extensions software is designed to be configured by two people with distinct responsibilities. This task division can be enforced by roles. Because discrete roles and additional 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.

Preparing an Oracle Solaris System and Adding Trusted Extensions

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

- To properly support Trusted Extensions, you must install the underlying Oracle Solaris OS securely. For Oracle Solaris installation choices that affect Trusted Extensions, see "Install an Oracle Solaris System Securely" on page 44.
- If you have been using the Oracle Solaris OS, check your current configuration against the requirements for Trusted Extensions. For factors that affect Trusted Extensions, see "Prepare an Installed Oracle Solaris System for Trusted Extensions" on page 44.

▼ Install an Oracle Solaris System Securely

This task applies to fresh installations of Oracle Solaris. If you are upgrading, see "Prepare an Installed Oracle Solaris System for Trusted Extensions" on page 44.

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

In Trusted Extensions, you use the `root` role, as well as roles that you create, to configure the system.

2 When you first log in to Oracle Solaris, assign a password to the `root` role account.

a. Open a terminal window.

b. Assume the root role.

At the prompt, provide a password that is different from your user account password.

```
% su -
Your password has expired. Create a new password.
Enter new password:      Type a password for root
Retype the password:      Retype the root password
#
```

Assign a password of at least six alphanumeric characters. The password must be difficult to guess, thus reducing the chance of an adversary gaining unauthorized access by attempting to guess passwords.

Next Steps Continue with "Add Trusted Extensions Packages to an Oracle Solaris System" on page 45.

▼ Prepare an Installed Oracle Solaris System for Trusted Extensions

This task applies to Oracle Solaris systems that have been in use, and on which you plan to run Trusted Extensions.

Before You Begin You must be in the `root` role in the global zone.

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

The Trusted Extensions labeled brand is an exclusive brand of zones. Refer to the `brands(5)` and `trusted_extensions(5)` man pages.

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

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. Oracle Solaris provides protection against insecure passwords, but cannot prevent a user from disclosing her or his password, or from writing it down.

Next Steps Continue with "Add Trusted Extensions Packages to an Oracle Solaris System" on page 45.

▼ Add Trusted Extensions Packages to an Oracle Solaris System

Before You Begin You have completed either "Prepare an Installed Oracle Solaris System for Trusted Extensions" on page 44 or "Install an Oracle Solaris System Securely" on page 44.

You must be assigned the Software Installation rights profile.

1 After logging in as the initial user, assume the `root` role in a terminal window.

```
% su -
Enter Password:      Type root password
#
```

2 Download and install the Trusted Extensions package.

Use either the command line or the Package Manager GUI.

- **In the terminal window, use the `pkg install` command.**

  ```
  $ pkg install system/trusted/trusted-extensions
  ```

  To install trusted locales, specify the short name for the locale. For example, the following command installs the Japanese locale:

  ```
  $ pkg install system/trusted/locale/ja &
  ```

- **In the terminal window, start the Package Manager GUI.**

  ```
  $ packagemanager &
  ```

  a. **Select the Trusted Extensions packages.**

     i. **Show the categories in the Desktop (GNOME) category.**

     ii. **Select the Trusted Extensions category.**

iii. **In the list of packages, click the checkbox for trusted-extensions.**

iv. **(Optional) In the list of packages, click the checkbox for any locales that you want to install.**

b. **To add the packages, click the Install/Update icon.**

Resolving Security Issues Before Enabling Trusted Extensions

For each system on which Trusted Extensions will be configured, you need to make some configuration decisions. For example, you need to decide whether to install the default Trusted Extensions configuration or customize your configuration.

▼ Secure System Hardware and Make Security Decisions Before Enabling Trusted Extensions

For each system on which 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 is performed on every Oracle Solaris system.

- For SPARC systems, choose a PROM security level and provide a password.
- For x86 systems, protect the BIOS.
- On all systems, protect `root` 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 Oracle supplies. Oracle also supplies other `label_encodings` files, which you can find in the `/etc/security/tsol` directory. The Oracle files are demonstration files. They might not be suitable for production systems.

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

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

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

Full Label Name	Proposed Zone Name
PUBLIC	public
CONFIDENTIAL: INTERNAL USE ONLY	internal
CONFIDENTIAL: NEED TO KNOW	needtoknow
CONFIDENTIAL : RESTRICTED	restricted

Note – The automatic configuration method creates the `public` and `internal` zones.

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 these roles early in the configuration process.

If you are not required to configure the system by using discrete roles, you can choose to configure the system in the `root` role. This method of configuration is less secure. The `root` role can perform all tasks on the system, while other roles typically perform a more limited set of tasks. Therefore, configuration is more controlled when being performed by the roles that you create.

5 **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 Trusted Extensions Service and Logging In

In the Oracle Solaris OS, 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.

Note – Your Trusted Extensions system does not require a network to run a desktop with a directly connected bitmapped display, such as a laptop or workstation. Network configuration is required to communicate with other systems.

▼ Enable Trusted Extensions and Reboot

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 "Preparing an Oracle Solaris System and Adding Trusted Extensions" on page 43 and "Resolving Security Issues Before Enabling Trusted Extensions" on page 46.

You must be in the `root` role in the global zone.

1 Move the panel from the top of the screen to the bottom of the screen.

Caution – If you fail to move the panel, you might be unable to reach the desktop's main menu or panels when you log in to Trusted Extensions.

a. In the top panel, right-click and select Properties.

b. Change the Orientation of the top panel to Bottom.

2 Open a terminal window and enable the `labeld` service.

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

The `labeld` service adds labels to the system and starts the device allocation services.

Caution – Do not perform other tasks on the system until the cursor returns to the prompt.

3 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.
```

Caution – If you are enabling and configuring Trusted Extensions remotely, carefully review Chapter 12, "Remote Administration in Trusted Extensions (Tasks)." Do not reboot until you have configured the system to allow remote administration. If you do not configure the Trusted Extensions system for remote administration, you will be unable to reach it from a remote system.

4 **Reboot the system.**

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

Next Steps Continue with "Log In to Trusted Extensions" on page 49.

▼ Log In to Trusted Extensions

Logging in places you in the global zone, which is an environment that recognizes and enforces mandatory access control (MAC).

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

Before You Begin You have completed "Enable Trusted Extensions and Reboot" on page 48.

1 **Log in by using the user account that you created during installation.**

In the login dialog box, type *username*, then type the 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 her or his 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 from disclosing her or his password or writing it down.

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

3 **Dismiss the dialog box that says that the label `PUBLIC` has no matching zone.**

You will create the zone after you assume the `root` role.

4 **Assume the `root` role.**

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

The `root` role appears in a pulldown menu.

b. Select the root role.

If prompted, create a new password for the role.

Note – You must log out 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.

Next Steps Continue with one of the following:

- To configure a default system, go to "Creating Labeled Zones" on page 55
- To customize your system before you create labeled zones, go to "Setting Up the Global Zone in Trusted Extensions" on page 51.
- If your system does not have a graphical display, go to Chapter 12, "Remote Administration in Trusted Extensions (Tasks)."

◆ ◆ ◆ CHAPTER 4

Configuring Trusted Extensions (Tasks)

This chapter covers how to configure Trusted Extensions on a system with a monitor. To work properly, Trusted Extensions software requires configuration of labels and zones. You can also configure network communications, roles, and users who can assume roles.

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

Setting Up the Global Zone in Trusted Extensions

To customize your Trusted Extensions configuration, perform the procedures in the following task map. To install the default configuration, go to "Creating Labeled Zones" on page 55.

Task	Description	For Instructions
Protect the hardware.	Protects hardware by requiring a password to change hardware settings.	"Controlling Access to System Hardware (Tasks)" in *Oracle Solaris Administration: 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.	"How to Check and Install Your Label Encodings File" on page 52
Enable an IPv6 network.	Enables IP to recognize labeled packets on an IPv6 network.	"How to Enable IPv6 Networking in Trusted Extensions" on page 54
Change the DOI.	Specifies a Domain of Interpretation (DOI) that is not 1.	"How to Configure the Domain of Interpretation" on page 55

Task	Description	For Instructions
Configure the LDAP server.	Configures a Trusted Extensions LDAP directory server.	Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)"
Configure LDAP clients.	Makes this system a client of the Trusted Extensions LDAP directory server.	"Make the Global Zone an LDAP Client in Trusted Extensions" on page 89

▼ How to 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 *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.

To edit the `label_encodings` file, you must be in the `root` role.

1. **Copy the `label_encodings` file to the disk.**

   To copy from portable media, see "How to Copy Files From Portable Media in Trusted Extensions" on page 77.

2. **In a terminal window, check the syntax of the file.**

   a. **Run the `chk_encodings` command.**

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

b. 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 *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 10/10/2010

---> 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.10.10.10
# cd /etc/security/tsol
# cp label_encodings label_encodings.tx.orig
# cp label.encodings.10.10.10 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 10/10/2010
```

Next Steps You must reboot the system before creating labeled zones.

▼ How to Enable IPv6 Networking in Trusted Extensions

Caution – The `txzonemgr` script does not support IPv6 address syntax. Therefore, you use the `tncfg` command to add IPv6 hosts to your Trusted Extensions network. For examples, see "Trusted Network Fallback Mechanism" on page 188 and Example 16–11.

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.

Before You Begin You are in the `root` role in the global zone.

- **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:
  - Verify that the entry is spelled correctly.
  - Verify that the system has been rebooted after adding the correct entry to the `/etc/system` file.
- If you add Trusted Extensions to an Oracle 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*

Next Steps You must reboot the system before creating labeled zones.

▼ How to 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 site does not use a DOI of `1`, you must modify the `doi` value in every security template.

Before You Begin You are in the `root` role in the global zone.

- **Specify your DOI value in the default security templates.**

```
# tncfg -t cipso set doi=n
# tncfg -t admin_low set doi=n
```

Note – Every security template must specify your DOI value.

See Also
- "Network Security Attributes in Trusted Extensions" on page 185
- "How to Create Security Templates" on page 203

Next Steps If you plan to use LDAP, go to Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)." You must configure LDAP before you create labeled zones.

Otherwise, continue with "Creating Labeled Zones" on page 55.

Creating Labeled Zones

The instructions in this section configure labeled zones. You have the option of creating two labeled zones automatically or manually creating zones.

Note – If you plan to use LDAP, go to Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)." You must configure LDAP before you create labeled zones.

Task	Description	For Instructions
1a. Create a default Trusted Extensions configuration.	The `txzonemgr -c` command creates two labeled zones from the `label_encodings` file.	"How to Create a Default Trusted Extensions System" on page 56
1b. Create a default Trusted Extensions configuration by using a GUI.	The `txzonemgr` script creates a GUI that presents the appropriate tasks as you configure your system.	"How to Create Labeled Zones Interactively" on page 57

Task	Description	For Instructions
1c. Manually step through zone creation.	The `txzonemgr` script creates a GUI that presents the appropriate tasks as you configure your system.	"How to Create Labeled Zones Interactively" on page 57
2. Create a working labeled environment.	In the default configuration, label two workspaces as `PUBLIC` and `INTERNAL USE ONLY`.	"How to Assign Labels to Two Zone Workspaces" on page 59
3. (Optional) Link to other systems on your network.	Configure labeled zone network interfaces and connect the global zone and labeled zones to other systems.	"Configuring the Network Interfaces in Trusted Extensions" on page 60

▼ How to Create a Default Trusted Extensions System

This procedure creates a working Trusted Extensions system with two labeled zones. Remote hosts have not been assigned to the system's security templates, so this system cannot communicate with any remote hosts.

Before You Begin You have completed "Log In to Trusted Extensions" on page 49. You have assumed the `root` role.

1 Open a terminal window in the fourth workspace.

2 (Optional) Review the `txzonemgr` man page.

```
# man txzonemgr
```

3 Create a default configuration.

```
# /usr/sbin/txzonemgr -c
```

This command copies the Oracle Solaris OS and Trusted Extensions software to a zone, creates a snapshot of the zone, labels the original zone, then uses the snapshot to create a second labeled zone. The zones are booted.

- The first labeled zone is based on the value of `Default User Sensitivity Label` in the `label_encodings` file.
- The second labeled zone is based on the value of `Default User Clearance` in the `label_encodings` file.

This step can take about 20 minutes. To install the zones, the script uses the root password from the global zone for the labeled zones.

Next Steps To use your Trusted Extensions configuration, go to "How to Assign Labels to Two Zone Workspaces" on page 59.

How to Create Labeled Zones Interactively

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. In this procedure, you create two labeled zones. If you are using the Trusted Extensions `label_encodings` file, you create the default Trusted Extensions configuration.

Before You Begin You have completed "Log In to Trusted Extensions" on page 49. You have assumed the `root` role.

You have not created a zone yet.

1 Run the `txzonemgr` command without any options.

```
# 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 configuration.

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. Or, you can click the Cancel button.

---

2 Install the zones by choosing one of the following methods:

- **To create two labeled zones, select `public` and `internal` zones from the dialog box.**
  - The first labeled zone is based on the value of `Default User Sensitivity Label` in the `label_encodings` file.
  - The second labeled zone is based on the value of `Default User Clearance` in the `label_encodings` file

  a. Answer the prompt to identify the system.

  If the `public` zone uses an exclusive IP stack, or if it has an IP address which is defined in DNS, use the hostname as defined in DNS. Otherwise, use the name of the system.

  b. Do not answer the prompt for a root password.

  The root password was set at system installation. The input to this prompt will fail.

  c. At the zone login prompt, type your user login and password.

  Then, verify that all services are configured by running the `svcs -x` command. If no messages display, all services are configured.

d. Log out of the zone and close the window.

Type exit at the prompt, and choose Close window from the Zone Console.

In another window, the installation of the second zone completes. This zone is built from a snapshot, so it builds quickly.

e. Log in to the second zone console and verify that all services are running.

```
# svcs -x
#
```

If no messages display, all services are configured. The Labeled Zone Manager is visible.

f. Double-click the internal zone in the Labeled Zone Manager.

Select Reboot, then click the Cancel button to return to the main screen. All zones are running. The unlabeled snapshot is not running.

- **To manually create zones, select Main Menu, and then, Create a Zone.**

  Follow the prompts. The GUI steps you through zone creation.

  After the zone is created and booted, you can return to the global zone to create more zones. These zones are created from a snapshot.

Example 4–2 Creating Another Labeled Zone

In this example, the administrator creates a restricted zone from the default `label_encodings` file.

First, the administrator opens the `txzonemgr` script in interactive mode.

```
# txzonemgr &
```

Then, the administrator navigates to the global zone and creates a zone with the name `restricted`.

```
Create a new zone:restricted
```

Then, the administrator applies the correct label.

```
Select label:CNF : RESTRICTED
```

From the list, the administrator selects the Clone option and then selects `snapshot` as the template for the new zone.

After the `restricted` zone is available, the administrator clicks Boot to boot the second zone.

To enable access to the `restricted` zone, the administrator changes the `Default User Clearance` value in the `label_encodings` file to `CNF RESTRICTED`.

How to Assign Labels to Two Zone Workspaces

This procedure creates two labeled workspaces and opens a labeled window in each labeled workspace. When this task is completed, you have a working, non-networked Trusted Extensions system.

Before You Begin You have completed either "How to Create a Default Trusted Extensions System" on page 56 or "How to Create Labeled Zones Interactively" on page 57.

You are the initial user.

1. **Create a `PUBLIC` workspace.**

   The label of the `PUBLIC` workspace corresponds to the `Default User Sensitivity Label`.

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

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

   c. **Select PUBLIC and click OK.**

2. **Provide your password at the prompt.**

   You are in a `PUBLIC` workspace.

3. **Open a terminal window.**

   The window is labeled `PUBLIC`.

4. **Create an INTERNAL USE ONLY workspace.**

   If you are using a site-specific `label_encodings` file, you are creating a workspace from the value of `Default User Clearance`.

   a. **Switch to the third workspace.**

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

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

5. **Provide your password at the prompt.**

   You are in an `INTERNAL` workspace.

6. **Open a terminal window.**

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

Your system is ready to use. You have two user workspaces and a role workspace. In this configuration, the labeled zones use the same IP address as the global zone to communicate with other systems. They can do so because, by default, they share the IP address as an `all-zones` interface.

Next Steps If you plan to have your Trusted Extensions system communicate with other systems, go to "Configuring the Network Interfaces in Trusted Extensions" on page 60.

Configuring the Network Interfaces in Trusted Extensions

Your Trusted Extensions system does not require a network to run a desktop with a directly connected bitmapped display, such as a laptop or workstation. However, network configuration is required to communicate with other systems. By using the `txzonemgr` GUI, you can easily configure the labeled zones and the global zone to connect to other systems. For a description of the configuration options for labeled zones, see "Access to Labeled Zones" on page 32. The following task map describes and links to network configuration tasks.

Task	Description	For Instructions
Configure a default system for regular users.	The system has one IP address and uses an `all-zones` interface to communicate between the labeled zones and the global zone. The same IP address is used to communicate with remote systems.	"How to Share a Single IP Address With All Zones" on page 61
Add an IP address to the global zone.	The system has more than one IP address and uses the global zone's exclusive IP address to reach a private subnet. The labeled zones cannot reach this subnet.	"How to Share a Single IP Address With All Zones" on page 61
Assign an IP address to every zone, where the zones share the IP stack.	The system has more than one IP address. In the simplest case, the zones share a physical interface.	"How to Add an IP Instance to a Labeled Zone" on page 61
Add an `all-zones` interface to the IP instance per zone.	The system can offer its labeled zones privileged services that are protected from remote attack.	"How to Add an IP Instance to a Labeled Zone" on page 61
Assign an IP address to every zone, where the IP stack is exclusive.	One IP address is assigned to every zone, including the global zone. A VNIC is created for each labeled zone.	"How to Add a Virtual Network Interface to a Labeled Zone" on page 62
Connect the zones to remote zones.	This task configures the network interfaces of the labeled zones and the global zone to reach remote systems at the same label.	"How to Connect a Trusted Extensions System to Other Trusted Extensions Systems" on page 63
Run a separate `nscd` daemon per zone.	In an environment where each subnet has its own name server, this task configures one `nscd` daemon per zone.	"How to Configure a Separate Name Service for Each Labeled Zone" on page 64

How to Share a Single IP Address With All Zones

This procedure enables every zone on the system to use one IP address, the IP address of the global zone, to reach other identically labeled zones or hosts. This configuration is the default. You must complete this procedure if you have configured the network interfaces differently, and want to return the system to the default network configuration.

Before You Begin You must be in the `root` role in the global zone.

1. **Run the `txzonemgr` command without any options.**

   ```
   # txzonemgr &
   ```

   The list of zones is displayed in the Labeled Zone Manager. For information about this GUI, see "How to Create Labeled Zones Interactively" on page 57.

2. **Double-click the global zone.**

3. **Double-click Configure Network Interfaces.**

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

   - Type of `phys`
   - IP address of your hostname
   - State of `up`

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

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

   All zones can use this shared IP address to communicate with remote systems at their label.

6. **Click Cancel to return to the zone command list.**

Next Steps To configure the system's external network, go to "How to Connect a Trusted Extensions System to Other Trusted Extensions Systems" on page 63.

How to Add an IP Instance to a Labeled Zone

This procedure is required if you use a shared IP stack and per zone addresses, and you plan to connect the labeled zones to labeled zones on other systems on the network.

In this procedure, you create an IP instance, that is, a per zone address, for one or more labeled zones. The labeled zones use their per-zone address to communicate with identically labeled zones on the network.

Before You Begin You must be in the `root` role in the global zone.

The list of zones is displayed in the Labeled Zone Manager. To open this GUI, see "How to Create Labeled Zones Interactively" on page 57. The labeled zone that you are configuring must be halted.

1. **In the Labeled Zone Manager, double-click a labeled zone to which to add an IP instance.**
2. **Double-click Configure Network Interfaces.**
   A list of configuration options is displayed.
3. **Select Add an IP instance.**
4. **If your system has more than one IP address, choose the entry with the desired interface.**
5. **For this labeled zone, supply an IP address and a prefix count.**
   For example, type `192.168.1.2/24`. If you do not append the prefix count, you are prompted for a netmask. The equivalent netmask for this example is `255.255.255.0`.
6. **Click OK.**
7. **To add a default router, double-click the entry that you just added.**
   At the prompt, type the IP address of the router, and click OK.

   Note – To remove or modify the default router, remove the entry, then create the IP instance again.
8. **Click Cancel to return to the zone command list.**

Next Steps To configure the system's external network, go to "How to Connect a Trusted Extensions System to Other Trusted Extensions Systems" on page 63.

▼ How to Add a Virtual Network Interface to a Labeled Zone

This procedure is required if you use an exclusive IP stack and per zone addresses, and you plan to connect the labeled zones to labeled zones on other systems on the network.

In this procedure, you create a VNIC and assign it to a labeled zone.

Before You Begin You must be in the `root` role in the global zone.

The list of zones is displayed in the Labeled Zone Manager. To open this GUI, see "How to Create Labeled Zones Interactively" on page 57. The labeled zone that you are configuring must be halted.

1. **In the Labeled Zone Manager, double-click the labeled zone to which you want to add a virtual interface.**

2. **Double-click Configure Network Interfaces.**

   A list of configuration options is displayed.

3. **Double-click Add a virtual interface (VNIC).**

   If your system has more than one VNIC card, more than one choice is displayed. Choose the entry with the desired interface.

4. **Assign a host name, or assign an IP address and a prefix count.**

   For example, type `192.168.1.2/24`. If you do not append the prefix count, you are prompted for a netmask. The equivalent netmask for this example is `255.255.255.0`.

5. **To add a default router, double-click the entry that you just added.**

   At the prompt, type the IP address of the router, and click OK.

   Note – To remove or modify the default router, remove the entry, then create the VNIC again.

6. **Click Cancel to return to the zone command list.**

   The VNIC entry is displayed. The system assigns the name *zonename_n*, as in `internal_0`.

Next Steps To configure the system's external network, go to "How to Connect a Trusted Extensions System to Other Trusted Extensions Systems" on page 63.

▼ How to Connect a Trusted Extensions System to Other Trusted Extensions Systems

In this procedure, you define your Trusted Extensions network by adding remote hosts to which your Trusted Extensions system can connect.

Before You Begin The Labeled Zone Manager is displayed. To open this GUI, see "How to Create Labeled Zones Interactively" on page 57. You are in the `root` role in the global zone.

1. **In the Labeled Zone Manager, double-click the global zone.**

2. **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.**
3. **Click Cancel to return to the zone command list.**
4. **In the Labeled Zone Manager, double-click a labeled zone.**
5. **Select Add Access to Remote Host.**
   a. **Type the IP address of the identically labeled zone on another Trusted Extensions system.**
   b. **Run the corresponding commands in the zone of the other Trusted Extensions system.**

See Also
- Chapter 15, "Trusted Networking (Overview)"
- "Labeling Hosts and Networks (Task Map)" on page 200

How to Configure a Separate Name Service for Each Labeled Zone

This procedure enables you to separately configure a name service daemon (`nscd`) in each labeled zone. 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.

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 naming server for that label.

Note – This configuration requires that you have advanced networking skills.

Before You Begin The Labeled Zone Manager is displayed. To open this GUI, see "How to Create Labeled Zones Interactively" on page 57. You are in the `root` role in the global zone.

1. **In the Labeled Zone Manager, select Configure per-zone name service, and click OK.**

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

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

   For assistance, see the `nscd`(1M) man page.

3 Reboot the system.

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

4 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 September 10, 2011  10:10:11 AM PDT
   See: nscd(1M)
   See: /var/svc/log/system-name-service-cache:default.log
Impact: None.
```

b. Verify the route to the subnetwork.

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

Example 4–3 Removing a Name Service Cache From Each Labeled Zone

After testing one name service daemon per zone, the system administrator decides to remove the name service daemons from the labeled zones and run the daemon in the global zone only. To return the system to the default name service configuration, the administrator opens the `txzonemgr` GUI, selects the global zone, and selects `Unconfigure per-zone name service`, then `OK`. This selection removes the `nscd` daemon in every labeled zone. Then, the administrator reboots the system.

Next Steps When configuring user and role accounts for each zone, you have three options.

- You can create LDAP accounts in a multilevel LDAP directory server.
- You can create LDAP accounts in separate LDAP directory servers, one server per label.
- You can create local accounts.

Separately configuring a name service daemon in each labeled zone has password implications for all users. Users must authenticate themselves to gain access to any of their labeled zones, including the zone that corresponds to their default label. Furthermore, either the administrator must create accounts locally in each zone, or the accounts must exist in an LDAP directory where the zone is an LDAP client.

In the special case where an account in the global zone is running the Labeled Zone Manager, `txzonemgr`, the account's information is copied into the labeled zones so that at least that account is able to log in to each zone. By default, this account is the initial user account.

Creating Roles and Users in Trusted Extensions

Role creation in Trusted Extensions is identical to role creation in Oracle Solaris. However, for an evaluated configuration, a Security Administrator role is required.

Task	Description	For Instructions
Create a security administrator role.	Creates a role to handle security-relevant tasks.	"How to Create the Security Administrator Role in Trusted Extensions" on page 66
Create a system administrator role.	Creates a role to handle system administration tasks that are not related to security.	"How to Create a System Administrator Role" on page 68
Create users to assume the administrative roles.	Creates one or more users who can assume roles.	"How to Create Users Who Can Assume Roles in Trusted Extensions" on page 68
Verify that the roles can perform their tasks.	Tests the roles.	"How to Verify That the Trusted Extensions Roles Work" on page 70
Enable users to log in to a labeled zone.	Starts the `zones` service so that regular users can log in.	"How to Enable Users to Log In to a Labeled Zone" on page 71

▼ How to Create the Security Administrator Role in Trusted Extensions

Before You Begin You are in the `root` role in the global zone.

1 To create the role, use the `roleadd` command.

For information about the command, see the `roleadd(1M)` man page.

Use the following information as a guide:

- Role name – `secadmin`
- `-c` Local Security Officer

  Do not provide proprietary information.
- `-m` *home-directory*
- `-u` *role-UID*
- `-S` *repository*
- `-K` *key=value*

  Assign the Information Security and User Security rights profiles.

Note – For all administrative roles, use the administrative labels for the label range, audit uses of the `pfexec` command, set `lock_after_retries=no`, and do not set password expiration dates.

```
# roleadd -c "Local Security Officer" -m \
-u 110 -K profiles="Information Security,User Security" -S files \
-K lock_after_retries=no \
-K min_label=ADMIN_LOW -K clearance=ADMIN_HIGH secadmin
```

2 Provide 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
#
```

Assign a password of at least six 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.

3 Use the Security Administrator role as a guide when you create other roles.

Possible roles include the following:

- admin Role – `System Administrator` rights profile
- oper Role – `Operator` rights profile

Example 4–4 Creating the Security Administrator Role in LDAP

After configuring the first system with a local Security Administrator role, the administrator creates the Security Administrator role in the LDAP repository. In this scenario, LDAP clients can be administered by the Security Administrator role that is defined in LDAP.

```
# roleadd -c "Site Security Officer" -d server1:/rpool/pool1/BayArea/secadmin
-u 111 -K profiles="Information Security,User Security" -S ldap \
-K lock_after_retries=no -K audit_flags=lo,ex:no \
-K min_label=ADMIN_LOW -K clearance=ADMIN_HIGH secadmin
```

The administrator provides an initial password for the role.

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

Next Steps To assign the local role to a local user, see "How to Create Users Who Can Assume Roles in Trusted Extensions" on page 68.

▼ How to Create a System Administrator Role

Before You Begin You are in the `root` role in the global zone.

1 Assign the System Administrator rights profile to the role.

```
# roleadd -c "Local System Administrator" -m -u 111  -K audit_flags=lo,ex:no\
-K profiles="System Administrator" -K lock_after_retries=no \
-K min_label=ADMIN_LOW -K clearance=ADMIN_HIGH sysadmin
```

2 Provide an initial password for the role.

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

▼ How to Create Users Who Can Assume Roles in Trusted Extensions

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 assigns the initial password, and the Security Administrator role assigns security-relevant attributes, such as a role.

Before You Begin You must be in the `root` role in the global zone. Or, if separation of duty is enforced, users who can assume the distinct roles of Security Administrator and System Administrator must be present to assume their roles and perform the appropriate steps in this procedure.

1 Create a user.

Either the `root` role or the System Administrator role performs this step.

Do not place proprietary information in the comment.

```
# useradd -c "Second User" -u 1201 -d /home/jdoe jdoe
```

2 After creating the user, modify the user's security attributes.

Either the `root` role or the Security Administrator role performs this step.

Note – For users who can assume roles, turn off account locking, and do not set password expiration dates. Also, audit uses of the `pfexec` command.

```
# usermod -K lock_after_retries=no -K idletime=5 -K idlecmd=lock \
-K audit_flags=lo,ex:no jdoe
```

Note – The values for `idletime` and `idlecmd` continue in effect when the user assumes a role. For more information, see "`policy.conf` File Defaults in Trusted Extensions" on page 128.

3 Assign a password of at least six alphanumeric characters.

```
# passwd jdoe
New Password:       Type password
Re-enter new Password:      Retype password
```

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.

4 Assign a role to the user.

The `root` role or the Security Administrator role performs this step.

```
# usermod -R oper jdoe
```

5 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, users can 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 140.

b. Customize user initialization files.

See "Customizing the User Environment for Security (Task Map)" on page 133.

c. Create multilevel copy and link files.

On a multilevel 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 130.

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

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

This user is going to have a label range that is wider than the default label range. So, the `root` role 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` role consults Table 1–2, and then creates the user. The administrator places the user's home directory in `/export/home1` rather than the default, `/export/home`.

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

Then, the `root` role assigns 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` role adds the Security Administrator role to the user's definition. The role was created in "How to Create the Security Administrator Role in Trusted Extensions" on page 66.

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

▼ How to Verify That the Trusted Extensions Roles Work

To verify each role, assume the role. Then, perform tasks that only that role can perform and attempt tasks that the role is not permitted to 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. **Assume the role.**

   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. **Test the role.**

   For the authorizations that are required to change user properties, see the `passwd(1)` man page.

- The System Administrator role should be able to create a user and modify user properties that require the `solaris.user.manage` authorization, such as the user's login shell. The System Administrator role should not be able to change user properties that require the `solaris.account.setpolicy` authorization.
- The Security Administrator role should be able to change user properties that require the `solaris.account.setpolicy` authorization. The Security Administrator should not be able to create a user or change a user's login shell.

▼ How to Enable Users to Log In to a Labeled Zone

When the system 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. After configuring the system, you rebooted. You can assume the `root` role.

1. **Log in and assume the `root` role.**
2. **Check the state of the zones service.**

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

3. **Restart the service.**

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

4. **Log out.**

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

Creating Centralized Home Directories in Trusted Extensions

In Trusted Extensions, users need access to their home directories at every label at which the users work. By default, home directories are created automatically by the automounter that is running in each zone. However, if you use an NFS server to centralize home directories, you must enable home directory access at every label for your users.

▼ How to Create the Home Directory Server in Trusted Extensions

Before You Begin You are in the `root` role in the global zone.

1. **Add Trusted Extensions software to the home directory server and configure its labeled zones.**
   - Because users require a home directory at every label that they they can log in to, create a home directory server at every user label. For example, if you create a default configuration, you would create a home directory server for the `PUBLIC` label and a server for the `INTERNAL` label.
2. **For every labeled zone, follow the automount procedure in "How to NFS Mount Files in a Labeled Zone" on page 178. Then, return to this procedure.**
3. **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.**
4. **Log out of the home directory server.**

▼ How to Enable Users to Access Their Remote Home Directories at Every Label by Logging In to Each NFS Server

In this procedure, you allow users to create a home directory at each label by letting them directly log in to each home directory server. After creating each home directory on the central server, users can access their home directories from any system.

Alternatively, you, as administrator, can create a mount point on each home directory server by running a script, then modifying the automounter. For this method, see "How to Enable Users to Access Their Remote Home Directories by Configuring the Automounter on Each Server" on page 73.

Before You Begin The home directory servers for your Trusted Extensions domain are configured.

- **Enable users to log in directly to each home directory server.**

  Typically, you have created one NFS server per label.

  a. **Instruct each user to log in to each NFS server at the label of the server.**

  b. **When the login is successful, instruct the user to log out of the server.**

     A home directory for the user is available at the label of the server when the login is successful.

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

     The home directory for their default label is available from the home directory server. 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.

Next Steps Users can log in at a different label from their default label by choosing a different label from the label builder during login.

▼ How to Enable Users to Access Their Remote Home Directories by Configuring the Automounter on Each Server

In this procedure you run a script that creates a mount point for home directories on each NFS server. Then, you modify the `auto_home` entry at the label of the server to add the mount point. Then, users can log in.

Before You Begin The home directory servers for your Trusted Extensions domain are configured as LDAP clients. User accounts have been created on the LDAP server by using the `useradd` command with the `-S ldap` option. You must be in the `root` role.

1. **Write a script that creates a home directory mount point for every user.**

   The sample script makes the following assumptions:

   - The LDAP server is a different server from the NFS home directory server.
   - The client systems are also different systems.
   - The `hostname` entry specifies the external IP address of the zone, that is, the NFS home directory server for its label.

- The script will be run on the NFS server in the zone that serves clients at that label.

```
#!/bin/sh
hostname=$(hostname)
scope=ldap

for j in $(getent passwd|tr ' ' _); do
    uid=$(echo $j|cut -d: -f3)
    if [ $uid -ge 100 ]; then
        home=$(echo $j|cut -d: -f6)
        if [[ $home == /home/* ]]; then
            user=$(echo $j|cut -d: -f1)
            echo Updating home directory for $user
            homedir=/export/home/$user
            usermod -md ${hostname}:$homedir -S $scope $user
            mp=$(mount -p|grep  " $homedir zfs" )
            dataset=$(echo $mp|cut -d" " -f1)
            if [[ -n $dataset ]]; then
                zfs set sharenfs=on $dataset
            fi
        fi
    fi
done
```

2 On each NFS server, run the preceding script in the labeled zone that serves clients at that label.

Troubleshooting Your Trusted Extensions Configuration

A misconfigured desktop can prevent use of the system.

▼ How to Move Desktop Panels to the Bottom of the Screen

Note – The default position for desktop panels is the top of the screen. However, in Trusted Extensions the trusted stripe covers the top of the screen. Thus, the panels must be on the side or at the bottom of the workspace. A default workspace has two desktop panels.

Before You Begin You must be in the `root` role to change the desktop panel location for the system.

1 If you have one visible desktop panel at the bottom of the screen, perform one of the following actions:

- **Use the right mouse button to add applets to the visible panel.**
- **Move the second, hidden desktop panel to the bottom of the screen by performing the following step.**

2. **Otherwise, create a bottom desktop panel for your login only, or for all users of the system.**

   - **To move the panels for your login only, edit the `top_panel_screen`*n* file in your home directory.**

     a. **Change to the directory that contains the file that defines the panel locations.**

     ```
     % cd $HOME/.gconf/apps/panel/toplevels
     % ls
     %gconf.xml    bottom_panel_screen0/   top_panel_screen0/
     % cd top_panel_screen0
     % ls
     %gconf.xml             top_panel_screen0/
     ```

     b. **Edit the `%gconf.xml` file, which defines the location of the top panels.**

     ```
     % vi %gconf.xml
     ```

     c. **Find all orientation lines, and replace the string `top` with `bottom`.**

     For example, make the orientation line appear similar to the following:

     ```
     /toplevels/orientation" type="string">
                    <stringvalue>bottom</stringvalue>
     ```

   - **To move the panels for all users of the system, modify the desktop configuration.**

     In a terminal window in the `root` role, perform the following commands:

     ```
     # export SETUPPANEL="/etc/gconf/schemas/panel-default-setup.entries"
     # export TMPPANEL="/tmp/panel-default-setup.entries"
     # sed 's/<string>top<\/string>/<string>bottom<\/string>/' $SETUPPANEL > $TMPPANEL
     # cp $TMPPANEL $SETUPPANEL
     # svcadm restart gconf-cache
     ```

3. **Log out of the system and log in again.**

   If you have more than one desktop panel, the panels stack at the bottom of the screen.

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 an Oracle 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, the `root` role might use portable media to transfer the `label_encodings` files to all systems. Label the media with `Trusted Path`.

Before You Begin To copy administrative files, you must be in the `root` 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 *Trusted Extensions User's Guide*.

   The File Browser displays the contents of the clean media.

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

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

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 *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 4–6 Keeping Configuration Files Identical on All Systems

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

For example, the administrator modifies the `policy.conf` file, and the default `login` and `passwd` files for this site. So, the administrator copies the following files to the permanent directory.

```
# mkdir /export/commonfiles
# cp  /etc/security/policy.conf \
# cp  /etc/default/login \
# cp  /etc/default/passwd \
# cp  /etc/security/tsol/label_encodings \
/export/commonfiles
```

The administrator uses the Device Manager to allocate a CD-ROM in the global zone, transfers the files to the CD, and affixes a `Trusted Path` label.

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 in the `root` role in the global zone.

1. **Allocate the appropriate device.**

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

   The 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/label_encodings /etc/security/tsol/label_encodings.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 *Trusted Extensions User's Guide*.

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

▼ How to Remove Trusted Extensions From the System

You must perform specific steps to remove the Trusted Extensions feature from an Oracle Solaris system.

Before You Begin You are in the `root` role in the global zone.

1 **Archive any data in the labeled zones that you want to keep.**

For portable media, affix a physical sticker with the sensitivity label of the zone to each archived zone.

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

For details, see "How to Remove a Non-Global Zone" in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

3 **Disable the Trusted Extensions service.**

```
# svcadm disable labeld
```

4 **Disable device allocation.**

```
# svcadm disable allocate
```

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

6 **Configure the system.**

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

◆ ◆ ◆ CHAPTER 5

Configuring LDAP for Trusted Extensions (Tasks)

This chapter covers how to configure the Oracle Directory Server Enterprise Edition (Directory Server) for use with Trusted Extensions. The Directory Server provides LDAP services. LDAP is the supported naming service for Trusted Extensions. The final section, "Creating a Trusted Extensions LDAP Client" on page 89, covers how to configure an LDAP client.

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.

To configure the LDAP server, follow the instructions in *one* of the following task maps:

- "Configuring LDAP on a Trusted Extensions Network (Task Map)" on page 79
- "Configuring an LDAP Proxy Server on a Trusted Extensions System (Task Map)" on page 80

Configuring LDAP on a Trusted Extensions Network (Task Map)

Task	Description	For Instructions
Set up a Trusted Extensions LDAP server.	If you do not have an existing Oracle Directory Server Enterprise Edition, make your first Trusted Extensions system the Directory Server. This system has no labeled zones. The other Trusted Extensions systems are clients of this server.	"Collect Information for the Directory Server for LDAP" on page 81 "Install the Oracle Directory Server Enterprise Edition" on page 82 "Configure the Logs for the Oracle Directory Server Enterprise Edition" on page 84
Add Trusted Extensions databases to the server.	Populate the LDAP server with data from the Trusted Extensions system files.	"Populate the Oracle Directory Server Enterprise Edition" on page 86

Task	Description	For Instructions
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 89

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

Use this task map if you have an existing Oracle Directory Server Enterprise Edition that is running on an Oracle 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 Oracle Directory Server Enterprise Edition" on page 86
Set up an LDAP proxy server.	Make one Trusted Extensions system the proxy server for the other Trusted Extensions systems. These other systems use this proxy server to reach the LDAP server.	"Create an LDAP Proxy Server" on page 88
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 Oracle Directory Server Enterprise Edition" on page 86
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 89

Configuring the Oracle Directory Server Enterprise Edition 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 an Oracle Directory Server Enterprise Edition (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, 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
Oracle Directory Server Enterprise Edition *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`.
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 Oracle Directory Server Enterprise Edition

The Directory Server packages are available from the Oracle web site for Sun Software Products (`http://www.oracle.com/us/sun/sun-products-map-075562.html`).

Before You Begin You are on a Trusted Extensions system with a global zone. The system has no labeled zones. You must be in the `root` role in the global zone.

Trusted Extensions LDAP servers are configured for clients that use `pam_unix` to authenticate to the LDAP repository. With `pam_unix`, the password operations, and therefore the password policy, are determined by the client. Specifically, the policy set by the LDAP server is not used. For the password parameters that you can set on the client, see "Managing Password Information" in *Oracle Solaris Administration: Security Services*. For information about `pam_unix`, see the `pam.conf(4)` man page.

Note – The use of `pam_ldap` on an LDAP client is not an evaluated configuration for Trusted Extensions.

---

1. **Before you install the Directory Server packages, 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:

   ```
   ## /etc/hosts
   ...
   192.168.5.5 myhost myhost.example-domain.com
   ```

2. **Download the Oracle Directory Server Enterprise Edition packages from the Oracle web site for Sun Software Products (`http://www.oracle.com/us/sun/sun-products-map-075562.html`).**

   Select the most recent software that is appropriate for your platform.

3. **Install the Directory Server packages.**

   Answer the questions by using the information from "Collect Information for the Directory Server for LDAP" on page 81. For a full list of questions, defaults, and suggested answers, see Chapter 11, "Setting Up Oracle Directory Server Enterprise Edition With LDAP Clients (Tasks)," in *Oracle Solaris Administration: Naming and Directory Services* and Chapter 12, "Setting Up LDAP Clients (Tasks)," in *Oracle Solaris Administration: Naming and Directory Services*.

4. **(Optional) Add the environment variables for the Directory Server to your path.**

   ```
   # $PATH
   /usr/sbin:.../opt/SUNWdsee/dsee6/bin:/opt/SUNWdsee/dscc6/bin:/opt/SUNWdsee/ds6/bin:
   /opt/SUNWdsee/dps6/bin
   ```

5 **(Optional) Add the Directory Server man pages to your `MANPATH`.**

```
/opt/SUNWdsee/dsee6/man
```

6 **Enable the `cacaoadm` program and verify that the program is enabled.**

```
# /usr/sbin/cacaoadm enable
# /usr/sbin/cacaoadm start
start: server (pid n) already running
```

7 **Ensure that the Directory Server starts at every boot.**

Templates for the SMF services for the Directory Server are in the Oracle Directory Server Enterprise Edition 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.

8 **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 *Oracle Solaris Administration: Naming and Directory Services*.

▼ Create an LDAP Client for the Directory Server

You use this client to populate your Directory Server for LDAP. You must perform this task before you populate the Directory Server.

You can create the client temporarily on the Trusted Extensions Directory Server, then remove the client on the server, or you can create an independent client.

Before You Begin You are in the `root` role in the global zone.

1 Add Trusted Extensions software to a system.

You can use the Trusted Extensions Directory Server, or add Trusted Extensions to a separate system.

2 On the client, configure LDAP in the `name-service/switch` service.

a. Display the current configuration.

```
# svccfg -s name-service/switch listprop config
config                      application
config/value_authorization  astring       solaris.smf.value.name-service.switch
config/default              astring       "files ldap"
config/host                 astring       "files dns"
config/netgroup             astring       ldap
config/printer              astring       "user files ldap"
```

b. Change the following property from the default:

```
# svccfg -s name-service/switch setprop config/host = astring: "files ldap dns"
```

3 In the global zone, run the `ldapclient init` command.

In this example, the LDAP client is in the `example-domain.com` domain. The server's IP address is `192.168.5.5`.

```
# ldapclient init -a domainName=example-domain.com -a profileName=default \
> -a proxyDN=cn=proxyagent,ou=profile,dc=example-domain,dc=com \
> -a proxyDN=cn=proxyPassword={NS1}ecc423aad0 192.168.5.5
System successfully configured
```

4 Set the server's `enableShadowUpdate` parameter to `TRUE`.

```
# ldapclient -v mod -a enableShadowUpdate=TRUE \
> -a adminDN=cn=admin,ou=profile,dc=example-domain,dc=com
System successfully configured
```

For information about the `enableShadowUpdate` parameter, see "enableShadowUpdate Switch" in *Oracle Solaris Administration: Naming and Directory Services* and the `ldapclient(1M)` man page.

▼ Configure the Logs for the Oracle Directory Server Enterprise Edition

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.

Before You Begin You must be in the `root` role in the global zone.

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 Oracle Directory Server Enterprise Edition

To work in Trusted Extensions, the server port of the Directory Server must be configured as a multilevel port (MLP) in the global zone.

Before You Begin You must be in the `root` role in the global zone.

1 **Start the `txzonemgr`.**

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

2 **Add a multilevel port for the TCP protocol to the global zone.**

The port number is 389.

3 **Add a multilevel port for the UDP protocol to the global zone.**

The port number is 389.

▼ Populate the Oracle Directory Server Enterprise Edition

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 You must be in the `root` role in the global zone. You are on an LDAP client where shadow updating is enabled. For the prerequisites, see "Create an LDAP Client for the Directory Server" on page 83.

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/tsol
# cp tnrhdb tnrhtp /setup/files
```

Caution – Do not copy the *`attr` files. Rather, use the `-S ldap` option to the commands that add users, roles, and rights profiles to the LDAP repository. These commands add entries for the `user_attr`, `auth_attr`, `exec_attr`, and `prof_attr` databases. For more information, see the `user_attr(4)` and `useradd(1M)` man pages.

3 Remove the `+auto_master` entry from the `/setup/files/auto_master` file.

4 Create the zone automaps in the staging area.

```
# cp /zone/public/root/etc/auto_home_public /setup/files
# cp /zone/internal/root/etc/auto_home_internal /setup/files
# cp /zone/needtoknow/root/etc/auto_home_needtoknow /setup/files
# cp /zone/restricted/root/etc/auto_home_restricted /setup/files
```

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/&
```

5 Use the `ldapaddent` command to populate the Directory Server with every file in the staging area.

For example, the following command populates the server from the `hosts` file in the staging area.

```
# /usr/sbin/ldapaddent -D "cn=directory manager" \
-w dirmgr123 -a simple -f /setup/files/hosts hosts
```

6 If you ran the `ldapclient` command on the Trusted Extensions Directory Server, disable the client on that system.

In the global zone, run the `ldapclient uninit` command. Use verbose output to verify that the system is no longer an LDAP client.

```
# ldapclient -v uninit
```

For more information, see the `ldapclient(1M)` man page.

7 **To populate the Trusted Extensions network databases in LDAP, use the `tncfg` command with the `-S ldap` option.**

For instructions, see "Labeling Hosts and Networks (Task Map)" on page 200.

Creating a Trusted Extensions Proxy for an Existing Oracle Directory Server Enterprise Edition

First, you need to add the Trusted Extensions databases to the existing Directory Server on an Oracle 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 populated the LDAP server from a client that was modified to set the `enableShadowUpdate` parameter to `TRUE`. For the requirement, see "Create an LDAP Client for the Directory Server" on page 83.

In addition, you have added the databases that contain Trusted Extensions information to the LDAP server from a client where the `enableShadowUpdate` parameter was set to `TRUE`. For details, see "Populate the Oracle Directory Server Enterprise Edition" on page 86.

You must be in the `root` role in the global zone.

1 **On a system that is configured with Trusted Extensions, create a proxy server.**

Note – You must run two `ldapclient` commands. After you run the `ldapclient init` command, you then run the `ldapclient modify` command to set the `enableShadowUpdate` parameter to `TRUE`.

The following are sample commands. The `ldapclient init` command defines proxy values.

```
# ldapclient init \
-a proxyDN=cn=proxyagent,ou=profile,dc=west,dc=example,dc=com \
-a domainName=west.example.com \
-a profileName=pit1 \
-a proxyPassword=test1234 192.168.0.1
System successfully configured
```

The `ldapclient mod` command enables shadow updating.

```
# ldapclient mod -a enableShadowUpdate=TRUE \
-a adminDN=cn=admin,ou=profile,dc=west,dc=example,dc=com \
-a adminPassword=admin-password
System successfully configured
```

For details, see Chapter 12, "Setting Up LDAP Clients (Tasks)," in *Oracle Solaris Administration: Naming and Directory Services*.

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 *Oracle Solaris Administration: Naming and Directory Services*.

Creating a Trusted Extensions LDAP Client

The following procedure creates an LDAP client for an existing Trusted Extensions Directory Server.

▼ Make the Global Zone an LDAP Client in Trusted Extensions

This procedure establishes the LDAP naming service configuration for the global zone on an LDAP client.

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 Oracle Directory Server Enterprise Edition, that is, the Directory Server, must exist. The server must be populated with Trusted Extensions databases, and this client system must be able to contact the server. So, the Directory Server must have assigned a security template to this client. A specific assignment is not required, a wildcard assignment is sufficient.

You must be in the `root` role in the global zone.

1 If you are using DNS, add `dns` to the `name-service/switch` configuration.

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

a. Display the current configuration.

```
# svccfg -s name-service/switch listprop config
config                      application
config/value_authorization  astring       solaris.smf.value.name-service.switch
config/default              astring       files ldap
config/netgroup             astring       ldap
config/printer              astring       "user files ldap"
```

b. Add `dns` to the `host` property and refresh the service.

```
# svccfg -s name-service/switch setprop config/host = astring: "files dns ldap"
# svccfg -s name-service/switch:default refresh
```

c. Verify the new configuration.

```
# svccfg -s name-service/switch listprop config
config                      application
config/value_authorization  astring       solaris.smf.value.name-service.switch
config/default              astring       files ldap
config/host                 astring       files dns ldap
config/netgroup             astring       ldap
config/printer              astring       "user files ldap"
```

The Trusted Extensions databases use the default configuration `files ldap`, so are not listed.

2 To create an LDAP client, run the `txzonemgr` command without any options.

```
# txzonemgr &
```

a. Double-click the global zone.

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 runs the `ldapclient init` command.

3 Complete client configuration by enabling shadow updates.

```
# ldapclient -v mod -a enableShadowUpdate=TRUE \
> -a adminDN=cn=admin,ou=profile,dc=domain,dc=suffix
System successfully configured
```

4 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, redo Step 2 through Step 4. 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.

PART II

Administration of Trusted Extensions

The chapters in this part describe how to administer Trusted Extensions.

Chapter 6, "Trusted Extensions Administration Concepts," introduces the Trusted Extensions feature.

Chapter 7, "Trusted Extensions Administration Tools," describes the administrative programs that are specific to Trusted Extensions.

Chapter 8, "Security Requirements on a Trusted Extensions System (Overview)," describes the required and configurable security requirements in Trusted Extensions.

Chapter 9, "Performing Common Tasks in Trusted Extensions (Tasks)," introduces Trusted Extensions administration.

Chapter 10, "Users, Rights, and Roles in Trusted Extensions (Overview)," introduces role-based access control (RBAC) in Trusted Extensions.

Chapter 11, "Managing Users, Rights, and Roles in Trusted Extensions (Tasks)," provides instructions on managing regular users of Trusted Extensions.

Chapter 12, "Remote Administration in Trusted Extensions (Tasks)," provides instructions on remotely administering Trusted Extensions.

Chapter 13, "Managing Zones in Trusted Extensions (Tasks)," provides instructions on managing labeled zones.

Chapter 14, "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 15, "Trusted Networking (Overview)," provides an overview of the network databases and routing in Trusted Extensions.

Chapter 16, "Managing Networks in Trusted Extensions (Tasks)," provides instructions on managing the network databases and routing in Trusted Extensions.

Chapter 18, "Multilevel Mail in Trusted Extensions (Overview)," describes mail-specific issues in Trusted Extensions.

Chapter 19, "Managing Labeled Printing (Tasks)," provides instructions on handling printing in Trusted Extensions.

Chapter 20, "Devices in Trusted Extensions (Overview)," describes the extensions Trusted Extensions provides to device protection in Oracle Solaris.

Chapter 21, "Managing Devices for Trusted Extensions (Tasks)," provides instructions on managing devices by using the Device Manager.

Chapter 22, "Trusted Extensions Auditing (Overview)," provides Trusted Extensions–specific information about auditing.

Chapter 23, "Software Management in Trusted Extensions (Reference)," describes how to administer applications on a Trusted Extensions system.

◆ ◆ ◆ CHAPTER 6

Trusted Extensions Administration Concepts

This chapter introduces you to administering a system that is configured with the Trusted Extensions feature.

Trusted Extensions and the Oracle Solaris OS

Trusted Extensions software adds labels to a system that is running the Oracle 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 Oracle Solaris OS

Trusted Extensions software uses rights profiles, roles, auditing, privileges, and other security features of Oracle Solaris. You can use Secure Shell, BART, the Cryptographic Framework, IPsec, and IP Filter with Trusted Extensions. All features of the ZFS file system are available in Trusted Extensions, including snapshots and encryption.

Differences Between Trusted Extensions and the Oracle Solaris OS

Trusted Extensions software extends the Oracle Solaris OS. The following list provides an overview. See also 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 Oracle 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 an Oracle 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 an Oracle Solaris system are enabled on a Trusted Extensions system. For example, many sites that configure Trusted Extensions require separation of duty when creating users and assigning security attributes.
- Trusted Extensions can change the default behavior of Oracle Solaris. For example, on a system that is configured with Trusted Extensions, device allocation is required.
- Trusted Extensions can narrow the options that are available in Oracle Solaris. For example, in Trusted Extensions, all zones are labeled zones. Unlike in Oracle Solaris, 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 multilevel version of the Oracle Solaris 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 Manager GUI to administer devices. In addition, the `updatehome` command is used to place startup files in users' home directories 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 Labeled Zone Manager is used to administer labeled zones, in addition to the `zonecfg` command.

- 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. By default, regular users cannot shut down the system.

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

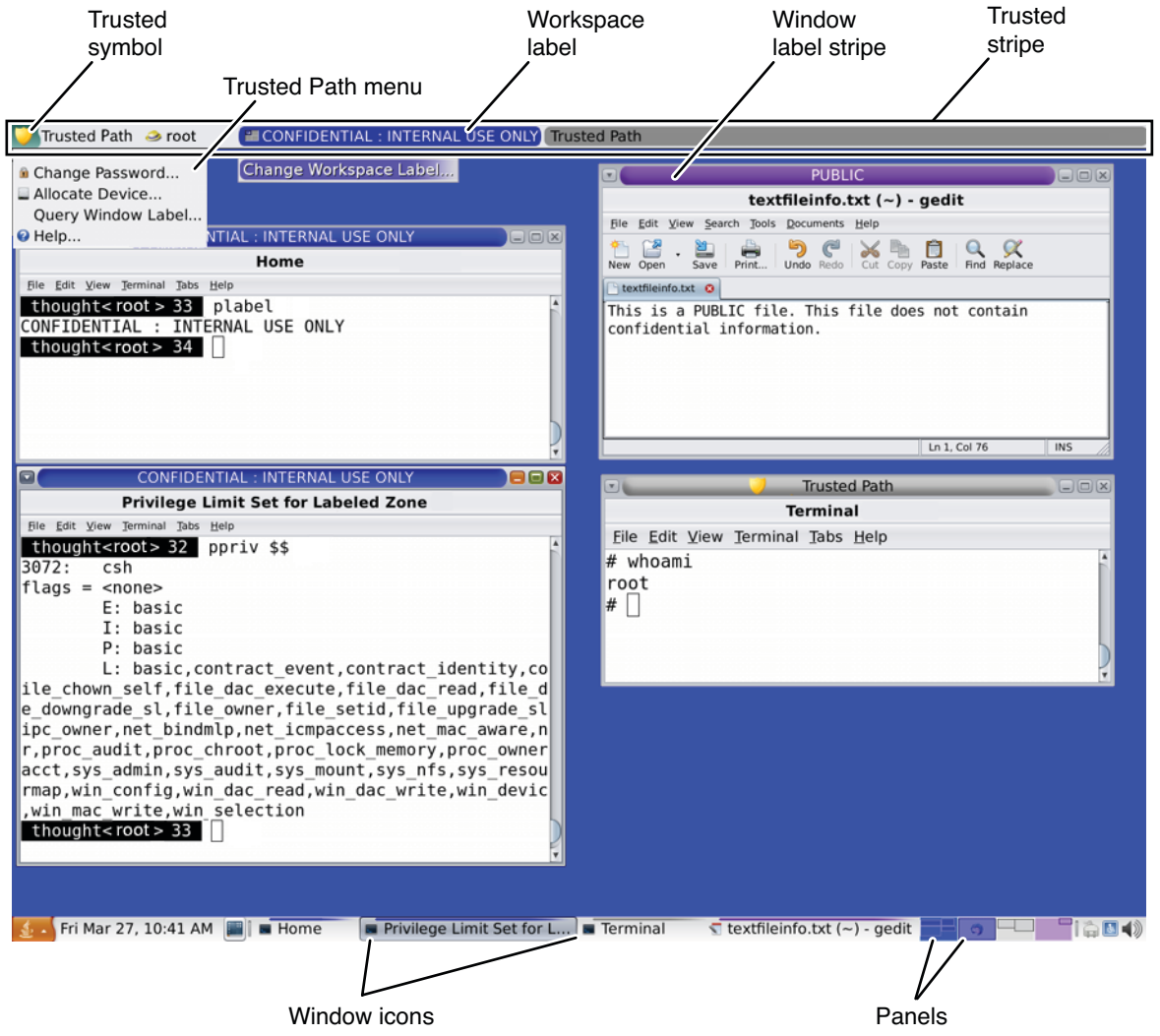
Trusted Extensions software adds labels to an Oracle Solaris system. Labeled workspaces and trusted applications, such as the Label Builder and the Device Manager, are also added. The concepts in this section are necessary to understand Trusted Extensions, both for users and administrators. Users are introduced to these concepts in the *Trusted Extensions User's Guide*.

Trusted Extensions Protections

Trusted Extensions software enhances the protection of the Oracle Solaris OS. Oracle Solaris 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, and 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 6–1.

FIGURE 6–1 Trusted Extensions Multilevel 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, such as when changing 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 Oracle Solaris provides. *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.

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` to `SECRET` to `PUBLIC`. 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 systems.

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 6–1 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`
`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 *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 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) are assigned security templates that describe their label range. Unlabeled hosts are also assigned a default label by the communicating Trusted Extensions system.
- **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 19, "Managing Labeled Printing (Tasks)," and "Labels on Printed Output" in *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 workspace panel and above window title bars, as shown in Figure 6–1.

  When a window is moved to a differently labeled workspace, the window maintains its original label. Processes that are initiated in that window execute at the 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.

Roles and Trusted Extensions

On a system that is running Oracle 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.

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.

As in Oracle Solaris, rights profiles are the basis of a role's capabilities. For information about rights profiles and roles, see Chapter 8, "Using Roles and Privileges (Overview)," in *Oracle Solaris Administration: Security Services*.

◆ ◆ ◆ CHAPTER 7

Trusted Extensions Administration Tools

This chapter describes the tools that are available in 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 Oracle 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 7–1 Trusted Extensions Administrative Tools

Tool	Description	For More Information
`/usr/sbin/txzonemgr`	Creates the Labeled Zone Manager GUI for creating and configuring labeled zones, including networking. Command-line options enable automatic creation of user-named zones.	See "Creating Labeled Zones" on page 55 and the `txzonemgr`(1M) man page. `txzonemgr` is a `zenity` (1) script.
Device Manager	Used to administer the label ranges of devices, and to allocate and deallocate devices.	See "Device Manager" on page 106 and "Handling Devices in Trusted Extensions (Task Map)" on page 255.

TABLE 7–1 Trusted Extensions Administrative Tools *(Continued)*

Tool	Description	For More Information
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" on page 139.
Selection Manager	Is also a tool for users who are authorized to change the security level of data. Appears when a program requires you to change the security level of data.	To authorize users, see "How to Enable a User to Change the Security Level of Data" on page 144. For an illustration, see "How to Move Data Between Labels" in *Trusted Extensions User's Guide*.
Trusted Extensions commands	Used to perform administrative tasks	For the list of administrative commands and configuration files, see Appendix D, "List of Trusted Extensions Man Pages."

`txzonemgr` Script

The `/usr/sbin/txzonemgr` command offers two modes.

- As a CLI, the command creates labeled zones from existing files. When run with the `-c` command option, the CLI creates and boots two labeled zones. The `-d` option deletes all labeled zones.
- As a GUI, the script displays a dialog box with the title Labeled Zone Manager. This GUI guides you through creating and booting labeled zones. The script includes cloning a zone to create a snapshot. Additionally, the GUI provides networking, naming service, and LDAP configuration menus.

The `txzonemgr` command runs a `zenity`(1) script. The Labeled Zone Manager dialog box 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.

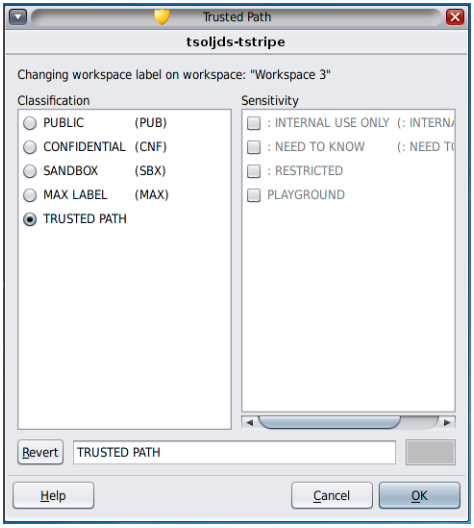
For more information about device protection in Trusted Extensions, see Chapter 21, "Managing Devices for Trusted Extensions (Tasks)."

Selection Manager in Trusted Extensions

The Selection Manager GUI appears when you attempt to change the label of an object or a selection. For more information, see "Rules When Changing the Level of Security for Data" on page 114.

Label Builder in Trusted Extensions

The label builder GUI supplies 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 2, "Logging In to Trusted Extensions (Tasks)," in *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. 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 under the `SENSITIVITY` section in the `label_encodings` file.

Developers can construct label builders for their applications by using the `tgnome-selectlabel` command. Type `tgnome-selectlabel -h` to display the online help. Also, see Chapter 6, "Label Builder GUI," in *Trusted Extensions Developer's Guide*.

Command Line Tools in Trusted Extensions

Commands that are unique to Trusted Extensions and commands that are modified by Trusted Extensions are contained in the *Oracle Solaris Reference Manual*. The `man` command finds all the commands. For a description of the commands, links to examples in the Trusted Extensions document set, and a link to the man pages, see Appendix D, "List of Trusted Extensions Man Pages."

Configuration Files in Trusted Extensions

The `/etc/inet/ike/config` file is extended by Trusted Extensions to include label information. The `ike.config(4)` man page describes 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.

◆ ◆ ◆ CHAPTER 8

Security Requirements on a Trusted Extensions System (Overview)

This chapter describes configurable security features on a system that is configured with Trusted Extensions.

Configurable Security Features

Trusted Extensions uses the same security features that Oracle Solaris provides, and adds some features. For example, the Oracle 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 Oracle Solaris in that you typically administer systems by assuming a role. As in the Oracle Solaris OS, configuration files are modified by the `root` role.

Roles in Trusted Extensions

In Trusted Extensions, roles are the conventional way to administer the system. Superuser is the `root` role, and is required for few tasks, such as setting audit flags, changing an account's password, and editing system files. Roles are created just as they are in Oracle Solaris.

The following roles are typical of a Trusted Extensions site:

- `root` **role** – Created at Oracle Solaris installation
- **Security Administrator role** – Created during or after initial configuration by the initial setup team
- **System Administrator role** – Created during or after initial configuration by the initial setup team

Role Creation in Trusted Extensions

To administer Trusted Extensions, you create roles that divide system and security functions.

The process of creating a role in Trusted Extensions is identical to the Oracle Solaris process. By default, roles are assigned the administrative label range of `ADMIN_HIGH` to `ADMIN_LOW`.

- For an overview of role creation, see "Using RBAC (Tasks)" in *Oracle Solaris Administration: Security Services*.
- To create roles, see "How to Create a Role" in *Oracle Solaris Administration: Security Services*.

Role Assumption in Trusted Extensions

On the trusted desktop, you can assume an assigned role by clicking your user name in the trusted stripe for the role choices. After confirming the role password, the current workspace is changed into a role workspace. A role workspace is in the global zone and has the trusted path attribute. Role workspaces are administrative workspaces.

Trusted Extensions Interfaces for Configuring Security Features

In Trusted Extensions, you can extend existing security features. Also, Trusted Extensions provides unique security features.

Extension of Oracle Solaris Security Features by Trusted Extensions

The following security mechanisms that Oracle Solaris provides are extensible in Trusted Extensions as they are in Oracle Solaris:

- **Audit classes** – Adding audit classes is described in Chapter 28, "Managing Auditing (Tasks)," in *Oracle Solaris Administration: Security Services*.

  Note – Vendors who want to add *audit events* need to contact an Oracle Solaris representative to reserve event numbers and obtain access to the audit interfaces.

- **Roles and rights profiles** – Adding roles and rights profiles is described in Chapter 9, "Using Role-Based Access Control (Tasks)," in *Oracle Solaris Administration: Security Services*.
- **Authorizations** – For an example of adding a new authorization, see "Customizing Device Authorizations in Trusted Extensions (Task Map)" on page 264.

As in Oracle Solaris, privileges cannot be extended.

Unique 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. Workspaces and their objects are labeled.
- **Device Manager –** By default, devices are protected by allocation requirements. The Device Manager GUI is the interface for administrators and for regular users.
- **Change Password menu –** This menu enables you to change your user or role password.
- **Change Workspace Label menu –** Users in multilevel sessions can change the workspace label. Users can be required to provide a password when entering a workspace of a different label.

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 provide additional suggestions.

Email Usage

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

Warn 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, 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 the `shadow(4)` man page.
- `auth_attr` **file** – Contains custom authorizations. See the `auth_attr(4)` man page.
- `prof_attr` **file** – Contains custom rights profiles . See the `prof_attr(4)` man page.
- `exec_attr` **file** – Contains commands with security attributes that the site has added to rights profiles. See the `exec_attr(4)` man page.
- **Audit trail** – Contains the audit records that the audit service has collected. See the `audit.log(4)` man page.

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 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 8–1 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 Browsers	Same label	Same UID	None
	Downgrade information	Same UID	`solaris.label.file.downgrade`
	Upgrade information	Same UID	`solaris.label.file.upgrade`
	Downgrade information	Different UIDs	`solaris.label.file.downgrade`
	Upgrade information	Different UIDs	`solaris.label.file.upgrade`

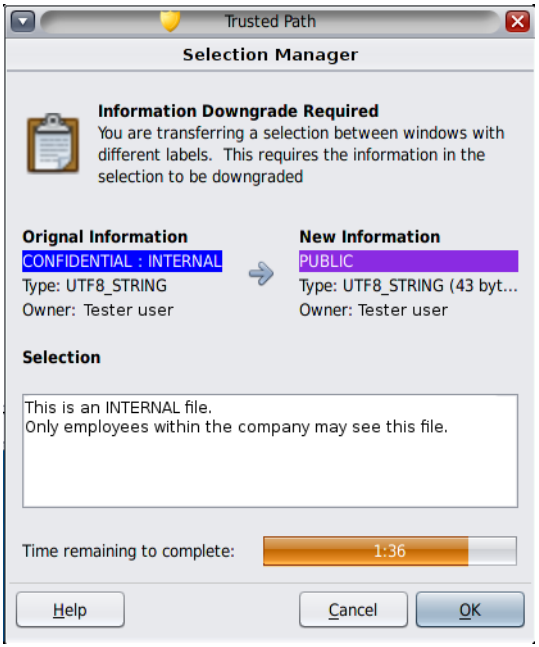
Different rules apply to selections within a window or file. Drag-and-drop of *selections* always requires equality of labels and ownership. Drag-and-drop between windows is mediated by the Selection Manager application, not by the `sel_config` file.

The rules for changing the label of selections are summarized in the following table.

TABLE 8–2 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 information	Same UID	`solaris.label.win.downgrade`
	Upgrade information	Same UID	`solaris.label.win.upgrade`
	Downgrade information	Different UIDs	`solaris.label.win.downgrade`
	Upgrade information	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, two-line, 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 `/usr/share/gnome/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 9

Performing Common Tasks in Trusted Extensions (Tasks)

This chapter introduces you to administering Trusted Extensions systems and contains tasks that are commonly performed on these systems.

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 to a Trusted Extensions system.	Logs you in securely.	"Logging In to Trusted Extensions" in *Trusted Extensions User's Guide*
Perform common user tasks on a desktop.	These tasks include: ■ Configuring your workspaces ■ Using workspaces at different labels ■ Using Trusted Extensions man pages	"Working on a Labeled System" in *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 *Trusted Extensions User's Guide*
Assume a role.	Places you in 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 118

Task	Description	For Instructions
Select a user workspace.	Exits you from the global zone.	"How to Exit the Global Zone in Trusted Extensions" on page 118

▼ 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.

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 138.

Before You Begin You are assigned an administrative role. For pointers, see "Role Creation in Trusted Extensions" on page 110.

1. **Click *account-name* in the trusted stripe.**

   From the list, select a role.

   For the location of Trusted Extensions desktop features, see Figure 6–1. For an explanation of these features, see Chapter 4, "Elements of Trusted Extensions (Reference)," in *Trusted Extensions User's Guide*.

2. **At the prompt, type the role password.**

   After authentication, the current workspace changes to the role workspace.

▼ How to Exit the Global Zone in Trusted Extensions

Before You Begin You are in the global zone.

1. **Select a user workspace from the desktop panel at the bottom of the screen.**

2. **Or, click your role name in the trusted stripe, and then select your user name.**

   The current workspace changes to a user workspace. All subsequent windows that you create in this workspace are created at your user label of the user.

   The windows that you created in the role workspace continue to support processes at the label of the role. Processes initiated in those windows execute in the global zone with administrative privileges

   For more information, see "Working on a Labeled System" in *Trusted Extensions User's Guide*.

Common Tasks in Trusted Extensions (Task Map)

The following task map describes common administrative procedures in Trusted Extensions.

Task	Description	For Instructions
Change the password for `root`.	Specifies a new password for the `root` role.	"How to Change the Password for `root`" on page 119
Reflect a password change in a labeled zone.	Reboots the zone to update the zone that a password has changed.	"How to Enforce a New Local User Password in a Labeled Zone" on page 120
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 120
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 121
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 123
Allocate a device.	Enables users to allocate devices. Uses a peripheral device to add information to or remove information from the system.	"How to Authorize Users to Allocate a Device" in *Oracle Solaris Administration: Security Services* "How to Allocate a Device in Trusted Extensions" in *Trusted Extensions User's Guide*
Administer a system remotely.	Administers Trusted Extensions systems from a remote system.	Chapter 12, "Remote Administration in Trusted Extensions (Tasks)"

▼ How to Change the Password for `root`

Trusted Extensions provides a GUI for changing your password.

1. **Assume the `root` role.**

   For the steps, see "How to Enter the Global Zone in Trusted Extensions" on page 118.

2. **Open the Trusted Path menu by clicking the trusted symbol in the trusted stripe.**

3. **Choose Change Login Password.**

   If separate passwords are created per zone, the menu can read Change Workspace Password.

4. **Change the password, and confirm the change.**

How to Enforce a New Local User Password in a Labeled Zone

Under the following conditions, labeled zones must be rebooted:

- One or more local users have changed their passwords.
- All zones are using a single instance of the naming service cache daemon (`nscd`).
- The system is administered with files, not LDAP.

Before You Begin You must be assigned the Zone Security rights profile.

- **To enforce the password change, reboot the labeled zones that the users can access.**

  Use one of the following methods:

  - **Use the `txzonemgr` GUI.**

    ```
    # txzonemgr &
    ```

    In the Labeled Zone Manager, navigate to the labeled zone and from the list of commands, select Halt, then select Boot.

  - **In a terminal window in the global zone, use zone administration commands.**

    You can choose to shut down or halt the system.

    - The `zlogin` command cleanly shuts down the zone.

      ```
      # zlogin labeled-zone shutdown -i 0
      # zoneadm -z labeled-zone boot
      ```

    - The `halt` subcommand bypasses the shutdown scripts.

      ```
      # zoneadm -z labeled-zone halt
      # zoneadm -z labeled-zone boot
      ```

Troubleshooting To automatically update user passwords for labeled zones, you must either configure LDAP or configure one naming service per zone. You can also configure both.

- To configure LDAP, see Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)."
- Configuring one naming service per zone requires advanced networking skills. For the procedure, see "How to Configure a Separate Name Service for Each Labeled Zone" on page 64.

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 9–1 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 9–2 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 118.

- **To obtain the 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 : INTERNAL USE ONLY"
    0x0004-08-48
    ```

    The string is not case-sensitive, but whitespace must be exact. For example, the following quoted strings return a hexadecimal label:

    - `"CONFIDENTIAL : INTERNAL USE ONLY"`
    - `"cnf : Internal"`
    - `"confidential : internal"`

    The following quoted strings return a parsing error:

    - `"confidential:internal"`
    - `"confidential: internal"`

  - **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 `label_encodings` file, the text equivalent of a clearance label does not include a colon (:).

Example 9–3 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

As in Oracle Solaris, in Trusted Extensions, the `root` account can change default security values on a system.

Files in the `/etc/security` and `/etc/default` directories contain security values. For more information, see Chapter 3, "Controlling Access to Systems (Tasks)," in *Oracle Solaris Administration: Security Services*.

Caution – Relax system security defaults only if site security policy allows you to.

Before You Begin You must be in the `root` role in the global zone.

- **Edit the system file.**

  The following table lists the security files and which security values you might change in the files.

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 *Oracle Solaris Administration: Security Services*. `passwd(1)` man page
`/etc/default/kbd`	Disable keyboard shutdown.	"How to Disable a System's Abort Sequence" in *Oracle Solaris Administration: 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. `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 10

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 Trusted Extensions. For details, see "Creating Roles and Users in Trusted Extensions" on page 66.

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 Oracle Solaris.

Administrator Responsibilities for Users

The System Administrator role creates user accounts. The Security Administrator role sets up the security aspects of an account.

For details on setting up users and roles, see the following:

- "Setting Up and Administering User Accounts (Task Map)" in *Oracle Solaris Administration: Common Tasks*
- Part III, "Roles, Rights Profiles, and Privileges," in *Oracle Solaris Administration: 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
- Assigning the initial password
- Modifying user and role properties, 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 the `root` 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 the actions that users can perform in Trusted Extensions and how much effort is required. Some decisions are the same as the decisions that you would make when installing the Oracle 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 originally configured by the initial setup team. For a description of the defaults, see "Default User Security Attributes in Trusted Extensions" on page 127.
- 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 136.
- Decide if users can access peripheral devices, such as the microphone, CD-ROM drive, and USB devices.

  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 268. To create a finer-grained set of device authorizations, see "Customizing Device Authorizations in Trusted Extensions (Task Map)" on page 264.

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 128.

`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 *Trusted Extensions Label Administration*.

Label values that the security administrator explicitly sets for individual users override values in the `label_encodings` file.

`policy.conf` File Defaults in Trusted Extensions

The `/etc/security/policy.conf` file contains the default security values for the system. Trusted Extensions adds two keywords to this file. To change the values system-wide, add these *keyword=value* pairs to the file. The following table shows the default values and the possible values for these keywords.

TABLE 10–1 Trusted Extensions Security Defaults in `policy.conf` File

Keyword	Default Value	Possible Values	Notes
`IDLECMD`	`LOCK`	`LOCK` | `LOGOUT`	Applies to the login user.
`IDLETIME`	`30`	`0` to `120` minutes	Applies to the login user.

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

For users who can log in at more than one label, you might want to set up two helper files, `.copy_files` and `.link_files`, in each user's minimum-label home directory. For more information, see "`.copy_files` and `.link_files` Files" on page 130.

Security Attributes That Must Be Assigned to Users

The security administrator can modify the security attributes for new users. For information about the files that contain the default values, see "Default User Security Attributes in Trusted Extensions" on page 127. The following table shows the security attributes that can be assigned to users and the effect of each assignment.

TABLE 10–2 Security Attributes That Are Assigned After User Creation

User Attribute	Location of Default Value	Is Action Required	Effect of Assignment
Password	None	Required	User has password
Roles	None	Optional	User can assume a role
Authorizations	`policy.conf` file	Optional	User has additional authorizations

TABLE 10–2 Security Attributes That Are Assigned After User Creation *(Continued)*

User Attribute	Location of Default Value	Is Action Required	Effect of Assignment
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	Kernel	Optional	User is audited differently from the system defaults

Security Attribute Assignment to Users in Trusted Extensions

The security administrator assigns security attributes to users 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, consider the following information:

Assigning Passwords
The system administrator can assign passwords to user accounts during account creation. After this initial assignment, the security administrator or the user can change the password.

As in Oracle Solaris, 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 user can be assigned more than one role if doing so is consistent with your site's security policy.

Assigning Authorizations
As in the Oracle Solaris OS, assigning authorizations to a user adds those authorizations to existing authorizations. Best practice is to add the authorizations to a rights profile, then assign the profile to the user.

Assigning Rights Profiles
As in the Oracle Solaris OS, the order of rights profiles is important. With the exception of authorizations, the profile mechanism uses the value of the first instance of an assigned security attribute. For more information, see "Order of Search for Assigned Security Attributes" in *Oracle Solaris Administration: Security Services*.

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 rights profile cannot 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, see "How to Restrict a User's Set of Privileges" on page 143.

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 Oracle Solaris OS, assigning audit classes to a user modifies the user's preselection mask. For more information about auditing, see Part VII, "Auditing in Oracle Solaris," in *Oracle Solaris Administration: Security Services* and Chapter 22, "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 136.

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                 .cshrc                  .mime_types
.aliases                .emacs                  .newsrc
.bashrc                 .login                  .signature
.bashrc.user            .mailrc                 .soffice
```

◆ ◆ ◆ CHAPTER 11

Managing Users, Rights, and Roles in Trusted Extensions (Tasks)

This chapter provides the 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. Many of these tasks are performed before regular users can log in.

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 134
Change Trusted Extensions policy for all users of a system.	Changes the `policy.conf` file.	"How to Modify `policy.conf` Defaults" on page 134
	Turns on the screensaver or logs out the user after a set amount of time that the system is idle.	Example 11–1
	Removes unnecessary privileges from all regular users of a system.	Example 11–2
	Prevents labels from appearing on printed output at a public kiosk.	Example 11–3
Configure initialization files for users.	Configures startup files, such as , `.bashrc`, `.cshrc`, `.copy_files`, and `.soffice` for all users.	"How to Configure Startup Files for Users in Trusted Extensions" on page 136

Task	Description	For Instructions
Log in to a failsafe session.	Fixes faulty user initialization files.	"How to Log In to a Failsafe Session in Trusted Extensions" on page 138

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 system.

Caution – You must complete this task before any regular users access the system.

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 118.

1. **Review the default user attribute settings in the `/etc/security/tsol/label_encodings` file.**
   For the defaults, see Table 1–2 in "Planning User Security in Trusted Extensions" on page 33.
2. **Modify the user attribute settings in the `label_encodings` file.**
3. **Distribute a copy of the file to every Trusted Extensions system.**

Caution – The `label_encodings` file must be the same on all systems. For one distribution method, see "How to Copy Files to Portable Media in Trusted Extensions" on page 75 and "How to Copy Files From Portable Media in Trusted Extensions" on page 77.

How to Modify `policy.conf` Defaults

Changing the `policy.conf` defaults in Trusted Extensions is identical to changing any security-relevant system file in Oracle Solaris. Use this procedure to change the defaults for all users of a system.

Before You Begin You must be in the `root` role in the global zone. For details, see "How to Enter the Global Zone in Trusted Extensions" on page 118.

1. **Review the default settings in the `/etc/security/policy.conf` file.**
   For Trusted Extensions keywords, see Table 10–1.
2. **Modify the settings.**

Example 11–1 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 `root` 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 `root` 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.

Note that if the login user assumes a role, the user's `IDLECMD` and `IDLETIME` values are in effect for that role.

Example 11–2 Modifying Every User's Basic Privilege Set

In this example, the security administrator of a large installation does not want regular users to view the processes of other users. Therefore, on every system that is configured with Trusted Extensions, the `root` role removes `proc_info` from the basic set of privileges. The `PRIV_DEFAULT` setting in the `/etc/policy.conf` file is uncommented and modified as follows:

```
PRIV_DEFAULT=basic,!proc_info
```

Example 11–3 Assigning Printing-Related Authorizations to All Users of a System

In this example, site security permits a public kiosk computer to print without labels. On the public kiosk, the `root` role modifies the value for `AUTHS_GRANTED` in the `/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. The administrator further modifies the `policy.conf` entry.

```
AUTHS_GRANTED=solaris.print.unlabeled,solaris.print.nobanner
```

After the public kiosk is rebooted, all print jobs are unlabeled, and have no banner or trailer pages.

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 118.

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. **In an editor, type the full pathname to the `.copy_files` file.**

      ```
      # vi /etc/skel/.copy_files
      ```

   b. **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 130 for ideas. For sample files, see Example 11–4.

3. **Customize the `.link_files` file.**

   a. **In an editor, type the full pathname to the `.link_files`.**

      ```
      # vi /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 which files to include in startup files, see "Customizing a User's Work Environment" in *Oracle Solaris Administration: Common Tasks*.
   - For details, see "How to Customize User Initialization Files" in *Oracle Solaris Administration: Common Tasks*.

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 11–4 Customizing Startup Files for Users

In this example, the system 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 system administrator creates a `.copy_files` and a `.link_files` file 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
.bashrc
.bashrc.user
.cshrc
.login
:wq

## .link_files for regular users with Korn shells
# Link these files to my home directory in every zone
.ksh
.profile
: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 customized files are copied to the appropriate skeleton directory.

```
$ cp .copy_files .link_files .bashrc .bashrc.user .cshrc \
.login .profile .mailrc /etc/skelC
```

```
$ cp .copy_files .link_files .ksh .profile .mailrc \
/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 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. **Type your user name in the login screen.**
2. **At the bottom of the screen, choose Solaris Trusted Extensions Failsafe Session from the desktop menu.**
3. **When prompted, type your password.**
4. **When prompted for an additional password, type the root password.**

   You can now debug the user's initialization files.

Managing Users and Rights (Task Map)

In Trusted Extensions, you assume the Security Administrator role to administer users, authorizations, rights, and roles. The following task map describes common tasks that you perform for users who operate in a labeled environment.

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" on page 139
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 140
Create a desktop that restricts a user to a few applications.	Assigns rights profiles that allow users to open only applications that appear on the desktop. The command line is not available, or optionally, accepts few commands.	"How to Limit a User to Desktop Applications" on page 141
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 143
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 143
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 144
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 144

▼ How to Modify a User's Label Range

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 view a list of the systems that run at a particular label. The user could view, but 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.

- **Do one of the following:**
  - **To extend the user's label range, assign a higher clearance.**

    ```
    # usermod -K min_label=INTERNAL -K clearance=ADMIN_HIGH jdoe
    ```

    You can also extend the user's label range by lowering the minimum label.

    ```
    # usermod -K min_label=PUBLIC -K clearance=INTERNAL jdoe
    ```

    For more information, see the `usermod(1M)` and `user_attr(4)` man pages.

- **To restrict the label range to one label, make the clearance equal to the minimum label.**

  ```
  # usermod -K min_label=INTERNAL -K clearance=INTERNAL jdoe
  ```

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 134.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **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 *Oracle Solaris Administration: Security Services*.

   The following authorizations that might be convenient for users:

   - `solaris.device.allocate` – Authorizes a user to allocate a peripheral device, such as a microphone or CD-ROM.

     By default, Oracle 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.
   - `solaris.label.file.downgrade` – Authorizes a user to lower the security level of a file
   - `solaris.label.file.upgrade` – Authorizes a user to heighten the security level of a file.
   - `solaris.label.win.downgrade` – Authorizes a user to select information from a higher-level file and place that information in a lower-level file.
   - `solaris.label.win.noview` – Authorizes a user to move information without viewing the information that is being moved.
   - `solaris.label.win.upgrade` – Authorizes a user to select information from a lower-level file and place that information in a higher-level file.
   - `solaris.login.remote` – Authorizes a user to remotely log in.
   - `solaris.print.ps` – Authorizes a user to print PostScript files.
   - `solaris.print.nobanner` - Authorizes a user to print hard copy without a banner page.
   - `solaris.print.unlabeled` – Authorizes a user to print hard copy that does not display labels.
   - `solaris.system.shutdown` – Authorizes a user to shut down the system and to shut down a zone.

2 **Assign the rights profile to a user or a role.**

For the step-by-step procedure, see "How to Change the RBAC Properties of a User" in *Oracle Solaris Administration: Security Services*.

▼ How to Limit a User to Desktop Applications

Site security might require that users have access only to applications that they can open from a desktop icon. This procedure assigns rights profiles that limit users to required applications only.

Note – On the Trusted Extensions desktop, the execution of commands is always based on rights profiles.

To enable every user of a particular system to be so authorized, see "How to Modify `policy.conf` Defaults" on page 134.

Before You Begin You must be in the Security Administrator role in the global zone.

1 **Create a rights profile called the Desktop applets rights profile that enables Oracle Solaris users to run the basic applets on their desktop.**

For the procedure, see "How to Restrict a User to Desktop Applications" in *Oracle Solaris Administration: Security Services*.

2 **Create another rights profile that enables Trusted Extensions users to run required trusted applets on their desktop.**

The lines are wrapped for display purposes.

```
# profiles -p "Trusted Desktop Applets"
profiles:Trusted Desktop Applets>
set desc="Can use trusted desktop applications except terminal"
profiles:Trusted Desktop Applets> add cmd=/usr/dt/config/tsoljds-migration;end
profiles:Trusted Desktop Applets> add cmd=/usr/bin/tsoljds-xagent;end
profiles:Trusted Desktop Applets> commit
```

3 **Add the Desktop Applets profile as a supplementary rights profile to the Trusted Desktop Applets profile.**

You created this rights profile in Step 2.

```
profiles:Trusted Desktop Applets> add profiles="Desktop Applets"
profiles:Trusted Desktop Applets> commit
profiles:Trusted Desktop Applets> exit
```

4 **Verify that the Trusted Desktop Applets rights profile contains the correct entries.**

Review the entries for errors, such as typos, omissions, or repetition.

```
# profiles -p "Trusted Desktop Applets" info
Found profile in files repository.
name=Trusted Desktop Applets
desc=Can use trusted desktop applications except terminal
profiles=Desktop Applets
cmd=/usr/dt/config/tsoljds-migration
cmd=/usr/bin/tsoljds-xagent
```

Tip – You can create a rights profile for an application or a class of applications that have desktop icons. Then, add the Trusted Desktop Applets rights profile as a supplementary rights profile for desktop access.

---

5 **Assign the user the `Trusted Desktop Applets` and `Stop` rights profiles.**

`# usermod -P "Trusted Desktop Applets,Stop"` *username*

This user can use the trusted desktop, but cannot launch a terminal window, act as the Console User, or have any of the rights contained in the Basic Solaris User rights profile.

Example 11–5 Enabling a Desktop User to Open a Terminal Window

In this example, the administrator enables a desktop user to open a terminal window. The administrator has already created the Desktop Applets rights profile for Oracle Solaris desktop users and the Trusted Desktop Applets rights profile for Trusted Extensions desktop users in the LDAP repository.

First, the administrator creates the Terminal Window rights profile and verifies its contents.

```
# profiles -p "Terminal Window" -S ldap
profiles:Terminal Window> set desc="Can open a terminal window"
profiles:Terminal Window> add cmd=/usr/bin/gnome-terminal;end
profiles:Terminal Window> commit
profiles:Terminal Window> exit
# profiles -p "Terminal Window" info
Found profile in ldap repository.
name=Terminal Window
desc=Can open a terminal window
cmd=/usr/bin/gnome-terminal
```

Then, the administrator assigns this rights profile and the All rights profile to desktop users who require terminal windows to perform their tasks. Without the All rights profile, users would not be able to run UNIX commands that do not require privilege, such as `ls` and `cat`.

```
# usermod -P "Trusted Desktop Applets,Terminal Window,All,Stop" -S ldap jdoe
```

With this set of rights profiles, user `jdoe` can use the desktop and terminal windows, but cannot act as the Console User or have any of the rights contained in the Basic Solaris User rights profile.

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.

Before You Begin You must be in the Security Administrator role in the global zone.

- **Remove one or more of the privileges in the `basic` set.**

Caution – Do not remove the `proc_fork` or the `proc_exec` privilege. Without these privileges, a user cannot use the system.

```
# usermod -K defaultpriv=basic,!proc_info,!proc_session,!file_link_any
```

By removing the `proc_info` privilege, you prevent the user from examining any processes that do not originate from the user. 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.

See Also For an example of collecting the privilege restrictions in a rights profile, see the examples following "How to Create or Change a Rights Profile" in *Oracle Solaris Administration: Security Services*.

To restrict the privileges of all users on a system, see Example 11–2.

How to Prevent Account Locking for Users

Perform this procedure for all users who can assume a role.

Before You Begin You must be in the Security Administrator role in the global zone.

- **Turn off account locking for a local user.**

```
# usermod -K lock_after_retries=no jdoe
```

To turn off account locking for an LDAP user, specify the LDAP repository.

```
# usermod -S ldap -K lock_after_retries=no jdoe
```

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 or of selected text. 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 168.

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 140 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. **Assign the profile to the appropriate users and roles.**

   For a step-by-step procedure, see "How to Change the RBAC Properties of a User" in *Oracle Solaris Administration: 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 in the global zone.

1. **Archive the user's home directory at every label.**
2. **Archive the user's mail files at every label.**
3. **Delete the user account.**

   ```
   # userdel -r jdoe
   ```

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.

For further considerations, see "User Deletion Practices" on page 113.

◆ ◆ ◆ CHAPTER 12

Remote Administration in Trusted Extensions (Tasks)

This chapter describes how to set up a Trusted Extensions system to be administered remotely, and how to log in and administer it.

Note – The configuration methods that headless and other 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.

Remote Administration in Trusted Extensions

Remote administration presents a significant security risk, particularly from users on untrusted systems. By default, Trusted Extensions does not allow remote administration from any system.

Until the network is configured, all remote hosts are assigned the `admin_low` security template, that is, they are recognized as unlabeled hosts. Until the labeled zones are configured, the only zone available is the global zone. In Trusted Extensions, the global zone is the administrative zone. Only a role can access it. Specifically, an account must have a label range from `ADMIN_LOW` to `ADMIN_HIGH` to reach the global zone.

While in this initial state, Trusted Extensions systems are protected from remote attacks by several mechanisms. Mechanisms include `netservices` values, default `ssh` policy, default login policy, and default PAM policy.

- At installation, no remote services except secure shell are enabled to listen on the network.

  However, the `ssh` service cannot be used for remote login by `root` or by role because of `ssh`, login, and PAM policies.

- The root account cannot be used for remote logins because root is a role. Roles cannot log in, as enforced by PAM.

  Even if root is changed to a user account, the default login and ssh policies prevent remote logins by the root user.
- Two default PAM values prevent remote logins.

  The pam_roles module rejects local and remote logins from accounts of type role.

  A Trusted Extensions PAM module, pam_tsol_account, 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.

So, as on an Oracle Solaris system, remote administration must be configured. Trusted Extensions adds two configuration requirements, the label range that is required to reach the global zone, and the pam_tsol_account module.

Methods for Administering Remote Systems in Trusted Extensions

In Trusted Extensions, you must use the ssh protocol with host-based authentication to reach and administer the remote system. Host-based authentication enables an identically-named user account to assume a role on the remote Trusted Extensions.

When host-based authentication is used, the ssh client sends both the original username and the role name to the remote system, the server. With this information, the server can pass sufficient content to the pam_roles module to enable role assumption without the user account logging in to the server.

The following methods of remote administration are possible in Trusted Extensions:

- **Administer from a Trusted Extensions system –** For the most secure remote administration, both systems assign their peer to a CIPSO security template. See Example 12–1.
- **Administer from an unlabeled system –** If administration by a Trusted Extensions system is not practical, the network protocol policy can be relaxed by specifying the allow_unlabeled option for the pam_tsol_account module in the pam.conf file.

  If this policy is relaxed, the default security template must be changed so that arbitrary systems cannot reach the global zone. The admin_low template should be used sparingly, and the wildcard address 0.0.0.0 must not default to the ADMIN_LOW label. For details, see "How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 212.

In either administrative scenario, to use the root role for remote login, you must relax PAM policy by specifying the allow_remote option for the pam_roles module.

Typically, administrators use the `ssh` command to administer remote systems from the command line. With the `-X` option, Trusted Extensions administrative GUIs can be used.

Also, you can configure the remote Trusted Extensions with the Xvnc server. Then, a Virtual Network Computing (VNC) connection can be used to display the remote multilevel desktop and administer the system. See "How to Configure a Trusted Extensions System With Xvnc for Remote Access" on page 152.

Configuring and Administering Remote Systems in Trusted Extensions (Task Map)

After enabling remote administration before rebooting the remote system into Trusted Extensions, you can configure the system by using Virtual Network Computing (VNC) or the `ssh` protocol.

Task	Description	For Instructions
Enable remote administration of a Trusted Extensions system.	Enables the administration of Trusted Extensions systems from specified `ssh` clients.	"Enable Remote Administration of a Remote Trusted Extensions System" on page 149
Enable Virtual Network Computing (VNC).	From any client, uses the Xvnc server on a remote Trusted Extensions system to display the server's multilevel session back to the client.	"How to Configure a Trusted Extensions System With Xvnc for Remote Access" on page 152
Log in remotely to a Trusted Extensions system.	Assumes a role on the remote system to administer it.	"How to Log In and Administer a Remote Trusted Extensions System" on page 153

Note – Consult your security policy to determine which methods of remote administration are permissible at your site.

▼ Enable Remote Administration of a Remote Trusted Extensions System

In this procedure, you enable host-based authentication on an Oracle Solaris remote system before adding the Trusted Extensions feature to it. The remote system is the `ssh` server.

Before You Begin The remote system is installed with Oracle Solaris and you can access that system.

1 **On both systems, enable host-based authentication.**

For the procedure, see "How to Set Up Host-Based Authentication for Secure Shell" in *Oracle Solaris Administration: Security Services*.

Note – Do not use the `cat` command. Copy and paste the public key over an `ssh` connection. If your ssh client is not an Oracle Solaris system, follow your platform's instructions for configuring an `ssh` client with host-based authentication.

After completing this step, you have a user account on both systems that can assume the `root` role. The accounts are assigned the same UID, GID, and role assignment. You also have generated public/private key pairs and have shared public keys.

2 **On the `ssh` server, relax `ssh` policy to enable `root` to log in remotely.**

```
# vi /etc/ssh/sshd_config
## Permit remote login by root
PermitRootLogin yes
```

A later step limits the `root` login to a particular system and user.

Note – Because the administrator is going to assume the `root` role, you do not need to relax the login policy that prevents remote `root` login.

---

3 **On the `ssh` server, restart the `ssh` service.**

```
# svcadm restart ssh
```

4 **On the `ssh` server, in `root`'s home directory, specify the host and user for host-based authentication.**

```
# cd
# vi .shosts
client-host username
```

The `.shosts` file enables *username* on the *client-host* system to assume the `root` role on the server, when a public/private key is shared.

5 **On the `ssh` server, relax the two PAM policies.**

a. **Allow remote login by roles.**

```
# vi /etc/pam.conf
...
# Default definition for Account management
# Used when service name is not explicitly mentioned for account management
#
# other   account requisite    pam_roles.so.1
# Enable remote role assumption
other   account requisite    pam_roles.so.1   allow_remote
...
```

This policy enables *username* on the *client-host* system to assume a role on the server.

b. Allow unlabeled hosts to contact the Trusted Extensions remote system.

```
# vi /etc/pam.conf
# Default definition for Account management
# Used when service name is not explicitly mentioned for account management
#
# other   account requisite    pam_roles.so.1
# Enable remote role assumption
other   account requisite    pam_roles.so.1   allow_remote
#
other   account required     pam_unix_account.so.1
# other   account required     pam_tsol_account.so.1
# Enable unlabeled access to TX system
other   account required     pam_tsol_account.so.1  allow_unlabeled
```

c. Copy your modified `pam.conf` file to `pam.conf.site`.

```
# cp /etc/pam.conf /etc/pam.conf.site
```

6 Test the configuration.

a. Open a new terminal on the remote system.

b. On *client-host*, in a window owned by *username*, assume the `root` role on the remote system.

```
% ssh -l root remote-system
```

7 After the configuration is proved to work, enable Trusted Extensions on the remote system and reboot.

```
# svcadm enable -s labeld
# /usr/sbin/reboot
```

Example 12–1 Assigning the CIPSO Host Type for Remote Administration

In this example, the administrator is using a Trusted Extensions system to configure a remote Trusted Extensions host. To do so, the administrator uses the `tncfg` command on each system to define the host type of the peer system.

remote-system `# tncfg -t cipso add host=192.168.1.12` *Client-host*

client-host `# tncfg -t cipso add host=192.168.1.22` *Remote system*

Because an unlabeled system can also configure the remote Trusted Extensions host, the administrator leaves the `allow_unlabeled` option in the remote host's `pam.conf` file.

Troubleshooting When the administrator upgrades to a new release of the Oracle Solaris OS, a new `pam.conf` file is not installed. For a description of the `preserve=true` file action on upgrade, see the `pkg`(5) man page.

How to Configure a Trusted Extensions System With Xvnc for Remote Access

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 server that is running Trusted Extensions, log in to the Xvnc server, then display and work on a multilevel desktop.

For more information, see the `Xvnc`(1) and `vncconfig`(1) man pages.

Before You Begin

You have installed and configured Trusted Extensions on this system that will be used as the Xvnc server. The global zone on this system has a fixed IP address, that is, it is not using the automatic network configuration profile, as described on the `netcfg`(1M)man page.

This system recognizes the VNC clients by hostname or by IP address. Specifically, the `admin_low` security template identifies either explicitly or by using a wildcard the systems that can be VNC clients of this server. For more information about configuring the connection securely, see "How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 212.

If you are currently running in a GNOME session on the console of the future Trusted Extensions Xvnc server, you do not have Desktop Sharing enabled.

You are in the `root` role in the global zone of the future Trusted Extensions Xvnc server.

1. **Load or update the Xvnc software.**

   ```
   # packagemanager &
   ```

   In the Package Manager GUI, search for "vnc" and choose from the available servers. One option is the TigerVNC X11/VNC server software.

2. **Enable the X Display Manager Control Protocol.**

   Modify the GNOME Display Manager (gdm) custom configuration file. In the `/etc/gdm/custom.conf` file, type `Enable=true` under the `[xdmcp]` heading,

   ```
   [xdmcp]
   Enable=true
   ```

3. **Insert the following line in the `/etc/gdm/Xsession` file around line 27.**

   ```
   DISPLAY=unix:$(echo $DISPLAY|sed -e s/::ffff://|cut -d: -f2)
   ```

4. **Enable the Xvnc server service.**

   ```
   # svcadm enable xvnc-inetd
   ```

5 **Log out all active GNOME sessions on this server.**

```
# svcadm restart gdm
```

Wait about one minute for the desktop manager to restart. Then, a VNC client can connect.

6 **Verify that the Xvnc software is enabled.**

```
# svcs | grep vnc
```

7 **On every VNC client of this Xvnc server, install the VNC client software.**

For the client system, you have a choice of software. You can use VNC software from the Oracle Solaris repository.

8 **To display the Xvnc server workspace on a VNC client, perform the following steps:**

a. **In a terminal window on the client, connect to the server.**

```
% /usr/bin/vncviewer Xvnc-server-hostname
```

For command options, see the `vncviewer`(1) man page.

b. **In the window that displays, type your user name and password.**

Continue with the login procedure. For a description of the remaining steps, see "Logging In to Trusted Extensions" in *Trusted Extensions User's Guide*.

▼ How to Log In and Administer a Remote Trusted Extensions System

This procedure enables you to use the command line and the `txzonemgr` GUI to administer a remote Trusted Extensions system.

Before You Begin The user, role, and role assignment are identically defined on the local and remote systems, as described in "Enable Remote Administration of a Remote Trusted Extensions System" on page 149.

1 **On the desktop system, enable processes from the remote system to display.**

```
desktop $ xhost + remote-sys
```

2 **Ensure that you are the user who is identically named on both systems.**

3 **From a terminal window, log in to the remote system.**

Use the `ssh` command to log in.

```
desktop $ ssh -X -l identical-username remote-sys
Password:       Type the user's password
remote-sys $
```

The -X option enables GUIs to display.

4 **In the same terminal window, assume the role that is defined identically on both systems.**

For example, assume the root role.

```
remote-sys $ su - root
Password:       Type the root password
```

You are now in the global zone. You can now use this terminal window to administer the remote system from the command line. GUIs will display on your screen. For an example, see Example 12–2.

Example 12–2 Configuring Labeled Zones on a Remote System

In this example, the administrator uses the txzonemgr GUI to configure labeled zones on a labeled remote system from a labeled desktop system. As in Oracle Solaris, the administrator enables X server access to the desktop system by using the -X option to the ssh command. The user jandoe is defined identically on both systems and can assume the role remoterole.

```
TXdesk1 $ xhost + TXnohead4

TXdesk1 $ ssh -X -l jandoe TXnohead4
Password: Ins1PwD1
TXnohead4 $
```

To reach the global zone, the administrator uses the jandoe account to assume the role remoterole. This role is defined identically on both systems.

```
TXnohead4 # su - remoterole
Password: abcd1EFG
```

In the same terminal, the administrator in the remoterole role starts the txzonemgr GUI.

```
TXnohead4 $ /usr/sbin/txzonemgr &
```

The Labeled Zone Manager runs on the remote system and displays on the local system.

Example 12–3 Logging In to a Remote Labeled Zone

The administrator wants to change a configuration file on a remote system at the PUBLIC label.

The administrator has two options.

- Remotely log in to the global zone, display the remote global zone, then change to the PUBLIC label, open a terminal window, and edit the file
- Remotely log in to the PUBLIC zone by using the ssh command from a PUBLIC terminal window and then edit the file

Note that if the remote system is running one naming service daemon (`nscd`) for all the zones, *and* the remote system is using the files naming service, the password for the remote zone is the password that was in effect when the zone was last booted. If the password for the remote `PUBLIC` zone was changed, but the zone was not booted after the change, the original password allows access.

Troubleshooting If the `-X` option does not work, you might need to install a package. X11 forwarding is disabled when the `xauth` binary is not installed. The following command loads the binary: **`pkg install pkg:/x11/session/xauth`**.

◆ ◆ ◆ CHAPTER 13

Managing Zones in Trusted Extensions (Tasks)

This chapter describes how non-global, or *labeled*, zones work on a Trusted Extensions system. Also included are procedures that are unique to labeled zones.

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. You 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 global zone is solely 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.

On a Trusted Extensions system, the file systems of a zone are usually mounted in the global zone 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.

Each zone is a discrete ZFS file system. Every zone can have 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 14, "Managing and Mounting Files in Trusted Extensions (Tasks)," and "Mounting Labeled ZFS Datasets" on page 174.

Zones in Trusted Extensions are built on the Oracle Solaris Zones product. For reference, see Part II, "Oracle Solaris Zones," in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

Zones and IP Addresses in Trusted Extensions

Your initial setup team assigned IP addresses to the global zone and the labeled zones. They considered three types of configurations as described in "Access to Labeled Zones" on page 32 and summarized as follows:

- The system has one IP address for the global zone and all labeled zones.

  This default configuration is useful on a system that uses DHCP software to obtain its IP address.
- 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 networked 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.

A fourth type of configuration for a non-global zone is available in Oracle Solaris, exclusive IP instances. In this configuration, a non-global zone is assigned its own IP instance and manages its own physical interfaces. Each zone operates as if it is a distinct system. For a description, see "Zone Network Interfaces" in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

If you configure exclusive IP instances in Trusted Extensions, each labeled zone operates as if it is a distinct *single-level* system. The multilevel networking features of Trusted Extensions rely

on features of a shared IP stack. This guide assumes 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 security template by communicating Trusted Extensions systems. The label range or set of labels in the security 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 label range of 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.
- In the default configuration, 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, see "How to Create a Multilevel Port for a Zone" on page 216.

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*. To share a labeled file system, see "How to Share File Systems From a Labeled Zone" on page 176.

Zone Administration Utilities in Trusted Extensions

Zone administration tasks can be performed from the command line. However, the simplest way to administer zones is to use the shell script, /usr/sbin/txzonemgr that Trusted Extensions provides. This script provides a menu-based wizard for creating, installing, initializing, and booting zones. txzonemgr uses the zenity command. For details, see the txzonemgr(1M) and zenity(1) man pages.

Managing Zones (Task Map)

The following task map describes zone management tasks that are specific to Trusted Extensions. The map also links to common procedures that are performed in Trusted Extensions just as they are performed on an Oracle 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 162
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 163
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 164

Task	Description	For Instructions
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 165
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 168
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 166.
Configure a new zone.	Creates a zone at a label that is not currently being used to label a zone on this system.	See "How to Create Labeled Zones Interactively" on page 57.
Create a multilevel port for an application.	Multilevel ports are useful for programs that require a multilevel feed into a labeled zone.	"How to Create a Multilevel Port for a Zone" on page 216 Example 16–19
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 179
Remove a labeled zone.	Completely removes a labeled zone from the system.	"How to Remove a Non-Global Zone" in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*

▼ How to Display Ready or Running Zones

Before You Begin You must be in the System Administrator role in the global zone.

1 Run the `txzonemgr &` command.

The zone names, their status, and their labels are displayed in a GUI.

2 Or, use the `zoneadm list -v` command

```
# zoneadm list -v
ID NAME       STATUS     PATH              BRAND       IP
 0 global     running    /                 ipkg        shared
 5 internal   running    /zone/internal    labeled     shared
 6 public     running    /zone/public      labeled     shared
```

The output does not list the labels of the zones.

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 In an editor, create the `getmounts` script.

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_HIGH
/dev:   ADMIN_HIGH
/system/contract:        ADMIN_HIGH
/proc:                   ADMIN_HIGH
/system/volatile:        ADMIN_HIGH
/system/object:          ADMIN_HIGH
/lib/libc.so.1:          ADMIN_HIGH
/dev/fd:        ADMIN_HIGH
/tmp:           ADMIN_HIGH
/etc/mnttab:    ADMIN_HIGH
/export:        ADMIN_HIGH
/export/home:   ADMIN_HIGH
/export/home/jdoe:   ADMIN_HIGH
/zone/public:        ADMIN_HIGH
/rpool:              ADMIN_HIGH
/zone:               ADMIN_HIGH
/home/jdoe:          ADMIN_HIGH
/zone/public:        ADMIN_HIGH
/zone/snapshot:      ADMIN_HIGH
/zone/internal:      ADMIN_HIGH
...
```

Example 13–1 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 sample 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/jdoe:   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
```

3 Start the zone.

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

Example 13–2 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 13–3 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 must first halt the zone. You must be in the root role in the global 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 Enable the dataset to be shared.

```
# zfs set sharenfs=on datasetdir/subdir
```

5 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. This step ensures that the dataset is not mounted before the zone is booted.

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 13–4 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
```

Then, the administrator shares the dataset.

```
# zfs set sharenfs=on 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 boots the `needtoknow` zone.

```
# zoneadm -z needtoknow boot
```

The dataset is now accessible.

Users in 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 The zone you plan to configure must be halted. You must be in the Security Administrator role in the global zone.

1. **Open the Labeled Zone Manager.**

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

2. **Configure the zone to enable relabeling.**

   a. **Double-click the zone.**

   b. **From the list, select Permit Relabeling.**

3. **Select Boot to restart the zone.**

4. **Click Cancel to return to the zone list.**

   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 144.

Example 13–5 Preventing Downgrades From the `internal` Zone

In this example, the security administrator wants to prevent the downgrade of `CNF: INTERNAL USE ONLY` files on a system that previously was used to downgrade files.

The administrator uses the Labeled Zone Manager to halt the `internal` zone, then selects Deny Relabeling from the `internal` zone menu.

◆ ◆ ◆ CHAPTER 14

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 Oracle Solaris. Because Trusted Extensions attaches a unique label to every non-global zone, all the files and file systems 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 Oracle Solaris mounts. The differences occur in the enforcement of MAC policy. Also, the `txzonemgr` script assumes that home directories are mounted as `/export/home`.

NFS shares in Trusted Extensions are similar to Oracle Solaris shares in a global zone. However, the sharing of 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 Oracle Solaris. For mounting files, the automounter, and the `mount` command can be used. For sharing files, the `sharenfs` property of ZFS datasets 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 Oracle Solaris. For mounting files, the automounter and the `mount` command can be used. In Trusted Extensions, a unique `auto_home_`*zone-name* 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 the ZFS share properties. For more discussion, see "Global Zone Processes and Labeled Zones" on page 160.

Labels affect which files can be mounted. Files are shared and mounted at a particular label.

- For a Trusted Extensions system to mount a file system on another Trusted Extensions system, the server and the client must have compatible remote host templates of type `cipso`.

  For a Trusted Extensions client to write to a file system that is NFS-mounted, the file system must be mounted with read-write permissions *and* be at the same label as the client.
- For a Trusted Extensions system to mount a file system from an unlabeled system, the single label that is assigned to the unlabeled system by the Trusted Extensions system must match the label of the Trusted Extensions system.

  Similarly, for a labeled zone to mount a file system from an unlabeled system, the single label that is assigned to the unlabeled system by the Trusted Extensions system must match the label of the labeled zone.
- File systems whose labels differ from the mounting zone and are mounted with LOFS can be viewed, but cannot be modified. For details on NFS mounts, see "Access to NFS Mounted File Systems in Trusted Extensions" on page 171.

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 171.

If site security forbids the viewing of lower-level objects, you can hide lower-level file systems from the user. For details, see "How to Disable the Mounting of Lower-Level Files" on page 165.

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 Oracle Solaris, a non-global zone can share file systems. Similarly, in Trusted Extensions, a labeled zone can share file systems. To share file systems from a labeled zone, you turn on the ZFS share properties of the file system.

When the status of the labeled zone is `ready` or `running`, the file system is shared at the label of the zone. For the procedure, see "How to Share File Systems From a Labeled Zone" on page 176.

Access to NFS Mounted File Systems in Trusted Extensions

To make lower-level directories that are NFS-mounted visible to users in a higher-level zone, requires the following preparation:

- **Server configuration** – On the NFS server, you export the ZFS file system by setting its share properties. For the procedure, see "How to Share File Systems From a Labeled Zone" on page 176.
- **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 178.

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 172.

Home directories are created when users are created. However, the home directories are created 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, 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 144.

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 "How to Enable Users to Access Their Remote Home Directories at Every Label by Logging In to Each NFS Server" on page 72.

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-level* map that is generated from a higher-level zone:

```
+auto_home_public
*    public-zone-IP-address:/export/home/&
```

The `txzonemgr` script sets up this `PUBLIC` entry in the `auto_master` file in the global zone:

```
+auto_master
/net        -hosts        -nosuid,nobrowse
/home        auto_home    -nobrowse
/zone/public/home      auto_home_public      -nobrowse
```

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

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 Oracle Solaris. 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 Example 16–19.

A Trusted Extensions system can also share its file systems with unlabeled hosts. A file system that is exported to an unlabeled host is *writable* if its label equals the label that is assigned to the remote host by the exporting zone. A file system that is exported to an unlabeled host is *readable* only if its label is dominated by the label that is assigned to the remote system.

Communication 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 their peer 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 for remote file systems is always NFS.

Mounting Labeled ZFS Datasets

ZFS provides a security label attribute, `mlslabel`, that contains the label of the data in the dataset. The `mlslabel` property is inheritable. When a ZFS dataset has an explicit label, the dataset cannot be mounted on an Oracle Solaris system that is not configured with Trusted Extensions.

If the `mlslabel` 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, that is, the `mlslabel` property is undefined, 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, unless the zone allows read-down mounts. If the zone allows read-down mounts, a lower-level file system 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.

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 file systems that are labeled.

Task	Description	For Instructions
Back up files.	Archives your data.	"How to Back Up Files in Trusted Extensions" on page 175
Restore data.	Restores data from a backup.	"How to Restore Files in Trusted Extensions" on page 175
Share a labeled file system.	Allows a labeled file system to be accessed by users on other systems.	"How to Share File Systems From a Labeled Zone" on page 176

Task	Description	For Instructions
Mount a file system that is shared by a labeled zone.	Allows the contents of a file system to be mounted read-write in a labeled zone at the same label. When a higher-level zone mounts the shared directory, the directory mounts read-only.	"How to NFS Mount Files in a Labeled Zone" on page 178
Create home directory mount points.	Creates mount points for every user at every label. This task enables users to access their home directory at every label on a system that is not the NFS home directory server.	"How to Enable Users to Access Their Remote Home Directories at Every Label by Logging In to Each NFS Server" on page 72
Hide lower-level information from a user who is working at a higher label.	Prevents the viewing of lower-level information from a higher level.	"How to Disable the Mounting of Lower-Level Files" on page 165
Troubleshoot file system mounting problems.	Resolves problems with mounting a file system.	"How to Troubleshoot Mount Failures in Trusted Extensions" on page 179

▼ How to Back Up Files in Trusted Extensions

Before You Begin You must be assigned the Media Backup rights profile. You are in the global zone.

- **For available methods, see "Sending and Receiving ZFS Data" in** *Oracle Solaris Administration: ZFS File Systems***.**

Caution – Only the following commands preserve labels.

- `/usr/lib/fs/ufs/ufsdump` for major backups
- `/usr/sbin/tar cT` for small backups
- A script calling either of these commands

  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

Before You Begin You are in the `root` role in the global zone.

- **For available methods, see "Sending and Receiving ZFS Data" in** *Oracle Solaris Administration: ZFS File Systems***.**

Caution – Only the following commands preserve labels.

- `/usr/lib/fs/ufs/ufsrestore` for major restores
- `/usr/sbin/tar xT` for small restores

For details on the `T` option to the `tar` command, see the `tar(1)` man page.

How to Share File Systems From a Labeled Zone

To mount or share directories that originate in labeled zones, set the appropriate ZFS share properties on the file system, 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 assigned the ZFS File System Management rights profile.

1. **Create a workspace at the label of the file system that is going to be shared.**

   For details, see "How to Add a Workspace at Your Minimum Label" in *Trusted Extensions User's Guide*.

2. **In the zone, create the file system.**

   ```
   # zfs create rpool/wdocs1
   ```

3. **Share the file system by setting ZFS share properties.**

   For example, the following set of commands shares a documentation file system for writers. The file system is shared read-write so that writers can modify their documents on this server. `setuid` programs are disallowed.

   ```
   # zfs set share=name=wdocs1,path=/wdocs1,prot=nfs,setuid=off,
   exec=off,devices=off rpool/wdocs1
   # zfs set sharenfs=on rpool/wdocs1
   ```

   The command line is wrapped for display purposes.

4 **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 file systems 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
  ```

5 **Display the file systems that are shared from your system.**

In the `root` role in the global zone, run the following command:

```
# zfs get all rpool
```

For more information, see "Querying ZFS File System Information" in *Oracle Solaris Administration: ZFS File Systems*

6 **To enable the client to mount the shared file system, see "How to NFS Mount Files in a Labeled Zone" on page 178.**

Example 14–1 Sharing the `/export/share` File System at the `PUBLIC` Label

For applications that run at the label `PUBLIC`, the system administrator enables users to read the documentation in the `/export/reference` file system of the `public` zone.

First, the administrator changes the workspace label to `public` workspace and opens a terminal window. In the window, the administrator sets selected `share` properties on the `/reference` file system. The following command is wrapped for display purposes.

```
# zfs set share=name=reference,path=/reference,prot=nfs,
setuid=off,exec=off,devices=off,rdonly=on rpool/wdocs1
```

Then, the administrator shares the file system.

```
# zfs set sharenfs=on rpool/reference
```

The administrator leaves the `public` workspace and returns to the Trusted Path workspace. Because users are not allowed to log in to this file server, the administrator shares the file system by putting the zone in the ready state:

```
# zoneadm -z public ready
```

Users can access the shared file system once it is 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. File systems from unlabeled and labeled hosts can be mounted on a Trusted Extensions labeled system. The system must have a route to the file server at the label of the mounting zone.

- To mount the files read-write from a single-label host, the assigned label of the remote host must match the label of the mounting zone. Two remote host configurations are possible.
  - The remote host is assigned the same label as the mounting zone.
  - The remote host is a multilevel server that includes the label of the mounting zone.
- File systems 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 Oracle Solaris:

- By default, file systems are mounted at boot.
- To mount file systems 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.

Before You Begin You must be on the client system, in the zone at the label of the files that you want to mount. Verify that the file system that you want to mount is shared. Unless you are using the automounter, you must be assigned the File System Management rights profile. To mount from lower-level servers, the zone on this client 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 Your Minimum Label" in *Trusted Extensions User's Guide*.

  - **Mount files dynamically.**

    In the labeled zone, use the `mount` command.

  - **Mount files when the zone boots.**

  - **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.**

▼ How to Troubleshoot Mount Failures in Trusted Extensions

Before You Begin You must be in the zone at the label of the file system that you want to mount. You must be the `root` role.

1 **Verify that the file systems on the NFS server are shared.**

2 **Check the security attributes of the NFS server.**

a. **Use the `tninfo` or `tncfg` command to find the IP address of the server or a range of IP addresses that includes the NFS server.**

The address might be directly assigned, or indirectly assigned through a wildcard mechanism. The address can be in a labeled or 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.

3 **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.

4 **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 15

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.

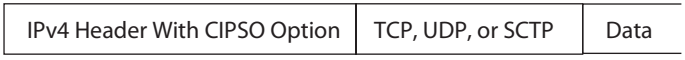
Typically on a trusted network, the label is generated by a sending host and processed by the receiving host. However, a trusted router can also add or strip labels while forwarding packets in a trusted network. A sensitivity label is mapped to a CIPSO label before transmission. The CIPSO label is embedded 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.

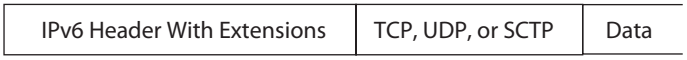
Trusted Extensions Data Packets

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.



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.



Trusted Network Communications

Trusted Extensions supports labeled and unlabeled hosts on a trusted network. The `txzonemgr` GUI and the `tncfg` command are used to configure the network.

Systems that run Trusted Extensions software support network communications between Trusted Extensions systems and any of the following types of hosts:

- Other hosts that are running Trusted Extensions
- Hosts that are running operating systems that do not recognize security attributes, but do support TCP/IP, such as Oracle Solaris systems, other UNIX systems, Microsoft Windows, and Macintosh OS systems
- Hosts that are running other trusted operating systems that recognize CIPSO labels

As in the Oracle Solaris OS, Trusted Extensions network communications and services can be managed by a naming service. Trusted Extensions adds the following interfaces to Oracle Solaris network interfaces:

- Trusted Extensions adds commands and provides a GUI to administer trusted networking. Trusted Extensions also adds options to the Oracle Solaris network commands. For a description of these commands, see "Network Commands in Trusted Extensions" on page 183.

  The interfaces manage three Trusted Extensions network configuration databases, `tnzonecfg`, `tnrhdb`, and `tnrhtp`. For details, see "Network Configuration Databases in Trusted Extensions" on page 184.
- Trusted Extensions adds the `tnrhtp` and `tnrhdb` databases to the properties of the naming service switch SMF service, `svc:/system/name-service/switch`.
- Part I, "Initial Configuration of Trusted Extensions," describes how to define zones and hosts when you configure the network. For additional procedures, see Chapter 16, "Managing Networks in Trusted Extensions (Tasks)."
- Trusted Extensions extends the IKE configuration file, `/etc/inet/ike/config`. For more information, see "Administration of Labeled IPsec" on page 195 and the `ike.config(4)` man page

Network Commands in Trusted Extensions

Trusted Extensions adds the following commands to administer trusted networking:

- `tncfg` – This command creates, modifies, and displays the configuration of your Trusted Extensions network. The `tncfg -t` command is used to view, create, or modify a specified security template. The `tncfg -z` command is used to view or modify the network properties of a specified zone. For details, see the `tncfg(1M)` man page.
- `tnchkdb` – This command is used to verify the correctness of the trusted network databases. The `tnchkdb` command is called whenever you change a security template (`tnrhtp`), a security template assignment (`tnrhdb`), or the configuration of a zone (`tnzonecfg`) by using the `txzonemgr` or the `tncfg` command. 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. For details, see the `tnctl(1M)` man page.
- `tnd` – This daemon pulls `tnrhdb` and `tnrhtp` information from the LDAP directory and local files. The order of search is dictated by the `name-service/switch` SMF service. The `tnd` daemon is started at boot time by the `svc:/network/tnd` service. This service is dependent on the `svc:/network/ldap/client`.

  In an LDAP network, the `tnd` 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 Oracle Solaris network commands:

- `ipadm` – The `all-zones` address property 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 `ipadm`(1M) man page.
- `netstat` – The `-R` option extends Oracle 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 Oracle 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 Oracle Solaris, 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.

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 host to host.

- `tnzonecfg` – This local database stores zone attributes that are security-related. The `tncfg` command is the interface to access and modify this database.

  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) man page. For a discussion of multilevel ports, see "Zones and Multilevel Ports" on page 159.
- `tnrhtp` – This database stores templates that describe the security attributes of hosts and gateways. The `tncfg` command is the interface to access and modify this database.

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 185.

- `tnrhdb` – This database holds the IP addresses and ranges of IP addresses that correspond to all hosts that are allowed to communicate with this system. The `tncfg` command is the interface to access and modify this database.

  Each host or range of IP addresses is assigned a security template from the `tnrhtp` database. The attributes in the template define the attributes of the assigned host.

Trusted Network Security Attributes

Network administration in Trusted Extensions is based on security templates. A security template describes a set of hosts that have identical protocols and security attributes.

Security attributes are administratively assigned to remote systems, both hosts and routers, by means of templates. The security administrator administers templates and assigns them to remote systems. If a remote system is not assigned a 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 186.
- A set of security attributes that are applied to each host type.

For more detail, see "Network Security Attributes in Trusted Extensions" on page 185.

Network Security Attributes in Trusted Extensions

A Trusted Extensions system is installed with a default set of security templates that are used to define the label properties of remote hosts. In Trusted Extensions, both unlabeled hosts and labeled hosts on the network are assigned security attributes by means of a security template. Hosts that are not assigned a template cannot communicate with hosts that are configured with Trusted Extensions. The templates are stored locally.

Hosts can be added to a security template by IP address or as part of a range of IP addresses. For further explanation, see "Trusted Network Fallback Mechanism" on page 188.

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 system or gateway.

  The Default label attribute is specific to the host type `unlabeled`. For details, see "Default Label in Security Templates" on page 187.
- **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.
- **Auxiliary 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 labels, hosts that are added to a template with an auxiliary label set can send and receive packets that match any one of the labels in the label set. The maximum number of auxiliary 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 Oracle 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 attributes might be too liberal 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 Oracle 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*. A technical reason to call these hosts "single-label" is that these hosts do not have `admin_high` and `admin_low` labels.

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 use this value in every security template, as described in "How to Configure the Domain of Interpretation" on page 55.

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 in the destination host's security template.
- 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.

Auxiliary Labels in Security Templates

The auxiliary 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 auxiliary label set is defined.

Trusted Network Fallback Mechanism

A host IP address can be added to a security template either directly or indirectly. Direct assignment adds a host's IP address. Indirect assignment adds a range of IP addresses that includes the host. To match a particular host, the trusted network software first looks for the specific IP address. If the search 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. For examples, see Table 15–1.

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 15–1 Trusted Extensions Host Address and Fallback Mechanism Entries

IP Version	Host Entry for `host_type=cipso`	IP Addresses Covered
IPv4	`192.168.118.57`	`192.168.118.57`
	`192.168.118.57/32`	The `/32` sets a prefix length of 32 fixed bits.
	`192.168.118.128/26`	From `192.168.118.0` through `192.168.118.63`
	`192.168.118.0`	All addresses on `192.168.118.` subnet.
	`192.168.118.0/24`	
	`192.168.0.0/24`	All addresses on `192.168.0.` subnet.
	`192.168.0.0`	All addresses on `192.168.` subnet.
	`192.168.0.0/16`	
	`192.0.0.0`	All addresses on `192.` subnet.
	`192.0.0.0/8`	
	`192.168.118.0/32`	Host address `192.168.118.0`. Not a range of addresses.
	`192.168.0.0/32`	Host address `192.168.0.0`. Not a range of addresses.
	`192.0.0.0/32`	Host address `192.0.0.0`. Not a range of addresses.
	`0.0.0.0/32`	Host address `0.0.0.0`. Not a range of addresses.
	`0.0.0.0`	All addresses on all networks
IPv6	`2001\:DB8\:22\:5000\:\:21f7`	`2001:DB8:22:5000::21f7`
	`2001\:DB8\:22\:5000\:\:0/52`	From `2001:DB8:22:5000::0` through `2001:DB8:22:5fff:ffff:ffff:ffff:ffff`
	`0\:\:0/0`	All addresses on all networks

Note that the `0.0.0.0/32` address matches the specific address, `0.0.0.0`. By adding the `0.0.0.0/32` entry to a system's unlabeled security template, you enable hosts with the specific address, `0.0.0.0`, to contact the system. 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 for an application that serves DHCP clients, see Example 16–16. The `0.0.0.0:admin_low` network is the default entry in the `admin_low` unlabeled host template. Review "How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 212 for security issues that would require changing this default.

For more information about prefix lengths in IPv4 and IPv6 addresses, see "Deciding on an IP Addressing Format for Your Network" in *Oracle Solaris Administration: 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 Oracle Solaris OS. Unlike Oracle Solaris, Trusted Extensions 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 Oracle Solaris routing makes choices to optimize the route. Trusted Extensions provides additional software to check security requirements that apply to the route choices. The Oracle 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 an auxiliary 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 hosts that support labels. When a label is not available from the packet, a default label is assigned to the message from the security template. 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 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.
- For all destinations, the label of the outgoing packet 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.
- 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 security template. Otherwise, the packet receives the indicated CIPSO label.
- Checks for forwarding a packet proceed similar to source accreditation:
  - 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.
  - For all destinations, the label of the outgoing packet 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.
  - 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 system 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 security template. 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. You can set up routes that enforce the degree of security that your site's security policy requires.

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 add the gateway's network interface that is connected to the external network to a single-label template . For more details about TCP/IP and routing, see the following:

- "Configuring an IPv4 Router" in *Oracle Solaris Administration: IP Services*
- "Configuring Component Systems on the Network" in *Oracle Solaris Administration: IP Services*
- "Major TCP/IP Administrative Tasks (Task Map)" in *Oracle Solaris Administration: IP Services*

- `netcfg(1M)`

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 routing tables are extended to include Trusted Extensions security attributes. The attributes are described in "Routing Table Entries in Trusted Extensions" on page 190. 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.

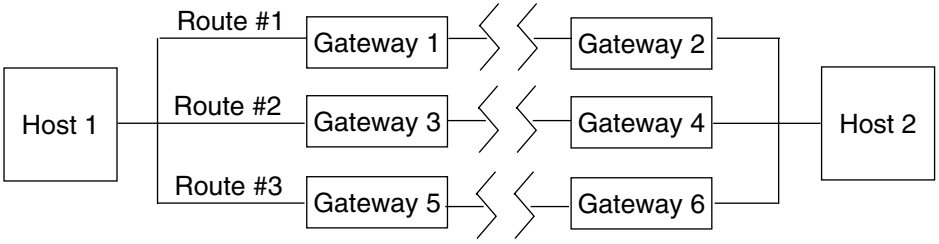
The routing software tries to find a route to the destination host in the routing tables. When the host is not explicitly named, the routing software looks for an entry for the subnet where the host resides. When neither the host nor the subnet is defined, the host sends the packet to a default gateway, 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.

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 15–1 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 Gateway 5's security template.

Routing Commands in Trusted Extensions

To display labels and extended security attributes for sockets, Trusted Extensions modifies the following Oracle 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.

To change routing table entries, Trusted Extensions provides the following interfaces:

- The `txzonemgr` GUI can be used to assign the default route for an interface.
- The `route -p` command with the `add` or `delete` option can be used to change routing table entries.

For examples, see "How to Add Default Routes" on page 215.

Administration of Labeled IPsec

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). Additionally, 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 *Oracle Solaris Administration: IP Services*. For Trusted Extensions modifications to IPsec procedures, see "Configuring Labeled IPsec (Task Map)" on page 218.

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 191.

The labels on IPsec packets from an application in a labeled zone 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 196, 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 191.

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 16

Managing Networks in Trusted Extensions (Tasks)

This chapter provides implementation details and procedures for securing a Trusted Extensions network.

Managing the Trusted Network (Task Map)

The following table links to the task maps for common Trusted Extensions networking procedures.

Task	Description	For Instructions
Assign labels to hosts and networks.	Creates remote host templates and assigns hosts to the security templates.	"Labeling Hosts and Networks (Task Map)" on page 200
Assign default routes and configure multilevel ports (MLPs).	Configures static routes that enable labeled packets to reach their destination through labeled and unlabeled gateways. Adds private and shared MLPs to labeled zones and the global zone.	"Configuring Routes and Multilevel Ports (Tasks)" on page 215
Enable IPsec to protect labeled packets.	Protects labeled packets with IPsec.	"Configuring Labeled IPsec (Task Map)" on page 218
Troubleshoot networking problems.	Steps to take when diagnosing network problems with labeled packets.	"Troubleshooting the Trusted Network (Task Map)" on page 223

Labeling Hosts and Networks (Task Map)

A Trusted Extensions system can contact other hosts only after the system has defined the security attributes of those hosts. Because remote hosts can have similar security attributes, Trusted Extensions provides security templates to which you can add hosts.

The following task map describes the tasks you can use to add to the security templates and to apply them to remote hosts.

Task	Description	For Instructions
View the security templates.	Displays the available security templates.	"How to View Security Templates" on page 201
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 202
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 206
Create security templates.	Creates security templates that define the security attributes of your trusted network.	"How to Create Security Templates" on page 203
	This security template changes the DOI to a value different from 1.	"How to Configure the Domain of Interpretation" on page 55
	This security template assigns a specific label to remote hosts.	Example 16–1
	Security template is for remote hosts that act as single-label gateways.	Example 16–2
	Security template is for remote hosts that restrict traffic to within a narrow label range.	Example 16–3
	Security template is for remote hosts with discrete labels.	Example 16–4
	Security template is for unlabeled remote hosts and networks.	Example 16–5
	Security template is for two remote hosts whose labels are disjoint from the rest of the network.	Example 16–6

Task	Description	For Instructions
Add a host to a security template.	Adds an IP address to a security template.	"How to Add a Host to a Security Template" on page 207 Example 16–7 Example 16–8 Example 16–9 Example 16–10
Add contiguous IP addresses to a security template.	Adds a range of IP addresses to a security template.	"How to Add a Range of Hosts to a Security Template" on page 210 Example 16–12 Example 16–13
Remove a host from a security template.	Removes the security definition of a host.	Example 16–11
Specify the hosts that can communicate at the `admin_low` label.	Increases security by specifying the hosts that a system can contact at boot time.	"How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 212
	Increases security by specifying a network of labeled hosts that the system can contact at boot time.	Example 16–14
Create an entry for the host address `0.0.0.0/32`.	Configures an application server to accept the initial contact from a remote client.	Example 16–16

How to View Security Templates

You can view the list of security templates and the contents of each template. The examples shown in this procedure are the default security templates.

1 List the available security templates.

```
# tncfg list
   cipso
   admin_low
```

2 View the contents of the listed templates.

```
# tncfg -t cipso info
   name=cipso
   host_type=cipso
   doi=1
   min_label=ADMIN_LOW
   max_label=ADMIN_HIGH
   host=127.0.0.1/32
```

The `127.0.0.1/32` entry in the preceding `cipso` security template identifies this system as labeled. When a peer assigns this system to the peer's remote host template with the `host_type` of `cipso`, the two systems can exchange labeled packets.

```
# tncfg -t admin_low info
   name=admin_low
   host_type=unlabeled
   doi=1
   def_label=ADMIN_LOW
   min_label=ADMIN_LOW
   max_label=ADMIN_HIGH
   host=0.0.0.0/0
```

The `0.0.0.0/0` entry in the preceding `admin_low` security template enables all hosts that are not explicitly assigned to a security template to contact this system. These hosts are recognized as unlabeled.

- The advantage of this entry is that all hosts that this system requires at boot time, such as servers and gateways, can be found.
- The disadvantage of this entry is that any host on this system's network can contact this system. To restrict which hosts can contact this system, see "How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 212.

▼ 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 security templates.

Follow the instructions in "How to View Security Templates" on page 201 to view the available security templates.

2 Create new security templates if you want to do any of the following for hosts that you communicate with:

- Limit the label range of a host or a group of hosts.
- Create a single-label host at a label other than `ADMIN_LOW`.
- Require a default label for unlabeled hosts that is not `ADMIN_LOW`.
- Create a host that recognizes a few discrete labels.
- Use a DOI other than `1`.

Next Steps To add hosts to the default security templates, go to "How to Add a Host to a Security Template" on page 207.

Otherwise, continue with "How to Create Security Templates" on page 203.

▼ How to Create Security Templates

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 values. The Security Administrator role includes these rights profiles.

1 (Optional) Determine the hexadecimal version of any label other than `ADMIN_HIGH` and `ADMIN_LOW`.

For labels such as `PUBLIC`, you can use either the label string or the hexadecimal value, `0x0002-08-08` as label values. The `tncfg` command accepts either format.

```
# atohexlabel "confidential : internal use only"
0x0004-08-48
```

For more information, see "How to Obtain the Hexadecimal Equivalent for a Label" on page 121.

2 Do not alter the default security templates.

For support purposes, do not delete the default security templates. You can copy and modify these templates. And you can add and remove hosts that are assigned to these templates. For an example, see "How to Limit the Hosts That Can Be Contacted on the Trusted Network" on page 212.

3 Create a security template.

The `tncfg -t` command provides three ways to create new templates.

- **Create a security template from scratch.**

  Use the `tncfg` command in interactive mode. The `info` subcommand displays the values that are supplied by default. Use the Tab key to complete partial properties and values.

  ```
  # tncfg -t newunlabeled
  tncfg:newtemplate> info
     name=newunlabeled
     host_type=unlabeled
     doi=1
     def_label=ADMIN_LOW
     min_label=ADMIN_LOW
     max_label=ADMIN_HIGH
  tncfg:newunlabeled> set m<Tab>
  set max_label=" set min_label="
  tncfg:newunlabeled> set ma<Tab>
  tncfg:newunlabeled> set max_label=ADMIN_LOW
  ...
  ```

  You can also supply the complete list of attributes for a security template on the command line. Semicolons separate the `set` subcommands. An omitted property receives the default value.

  ```
  # tncfg -t newunlabeled set host_type=unlabeled;set doi=1; \
  set min_label=ADMIN_LOW;set max_label=ADMIN_LOW
  ```

- **Copy and modify an existing security template.**

```
# tncfg -t cipso
tncfg:cipso> set name=newcipso
tncfg:newcipso> info
name=newcipso
host_type=cipso
doi=1
min_label=ADMIN_LOW
max_label=ADMIN_HIGH
```

  Hosts that are assigned to the existing security template are not copied to the new template.

- **Use a template file that the `export` subcommand creates.**

```
# tncfg -f unlab_1 -f template-file
tncfg: unlab1> set host_type=unlabeled
...
# tncfg -f template-file
```

  For an example of creating a source template for importing, see the `tncfg`(1M) man page..

Example 16–1 Creating a Security Template for a Gateway That Handles Packets at One Label

In this example, the security administrator defines a gateway that can only pass packets at the label `PUBLIC`.

```
# tncfg -t cipso_public
tncfg:cipso_public> set host_type=cipso
tncfg:cipso_public> set doi=1
tncfg:cipso_public> set min_label="public"
tncfg:cipso_public> set max_label="public"
tncfg:cipso_public> commit
tncfg:cipso_public> exit
```

The security administrator then adds the gateway host to the security template. For the addition, see Example 16–7.

Example 16–2 Creating a Security Template for an Unlabeled Router That Routes Labeled Packets

Any IP router can forward messages with CIPSO labels even though the router does not explicitly support labels. Such an unlabeled router requires a default label to define the level at which connections to the router, perhaps for router management, must 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`.

The security administrator creates the template from scratch.

```
# tncfg -t unl_public
tncfg:unl_public> set host_type=unlabeled
tncfg:unl_public> set doi=1
tncfg:unl_public> set def_label="PUBLIC"
tncfg:unl_public> set min_label=ADMIN_LOW
```

```
tncfg:unl_public> set max_label=ADMIN_HIGH
tncfg:unl_public> exit
```

The security administrator then adds the router to the security template. For the addition, see Example 16–8.

Example 16–3 Creating a Security Template for a Gateway With a Limited Label Range

In this example, the security administrator creates a gateway that restricts packets to a narrow label range. The administrator creates a security template and later assigns the gateway host to the security template.

```
# tncfg -t cipso_iuo_rstrct
tncfg:cipso_iuo_rstrct> set host_type=cipso
tncfg:cipso_iuo_rstrct> set doi=1
tncfg:cipso_iuo_rstrct> set min_label=0x0004-08-48
tncfg:cipso_iuo_rstrct> set max_label=0x0004-08-78
tncfg:cipso_iuo_rstrct> add host=192.168.131.78
tncfg:cipso_iuo_rstrct> exit
```

The security administrator then adds the gateway host to the security template. For the addition, see Example 16–9.

Example 16–4 Creating a Security Template That Has Discrete Labels

In this example, the security administrator creates a security template that recognizes two labels only, `confidential : internal use only` and `confidential : restricted`. All other traffic is rejected.

First, the administrator is careful to type the labels precisely. The software recognizes labels in uppercase and lowercase letters and by short name, but does not recognize labels where the spacing is inaccurate. For example, the label `confidential :restricted` is not a valid label.

```
# tncfg -t cipso_int_and_rst
tncfg:cipso_int_and_rst> set host_type=cipso
tncfg:cipso_int_and_rst> set doi=1
tncfg:cipso_int_and_rst> set min_label="cnf : internal use only"
tncfg:cipso_int_and_rst> set max_label="cnf : internal use only"
tncfg:cipso_int_and_rst> set aux_label="cnf : restricted"
tncfg:cipso_int_and_rst> exit
```

Example 16–5 Creating an Unlabeled Security Template at the Label `PUBLIC`

In this example, the security administrator creates a single-label template for hosts that can receive and send packets at the `PUBLIC` label only. This template might be assigned to hosts whose file systems must be mounted at the `PUBLIC` label by Trusted Extensions systems.

```
# tncfg -t public
tncfg:public> set host_type=unlabeled
tncfg:public> set doi=1
```

```
tncfg:public> set def_label="public"
tncfg:public> set min_sl="public"
tncfg:public> set max_sl="public"
tncfg:public> exit
```

The security administrator then adds hosts to the security template. For the addition, see Example 16–13.

Example 16–6 Creating a Labeled Security Template for Developers

In this example, the security administrator creates a `cipso_sandbox` template. This security template is assigned to systems that are used by developers of trusted software. Developer tests do not affect other labeled hosts because the label `SANDBOX` is disjoint from the other labels on the network.

```
# tncfg -t cipso_sandbox
tncfg:cipso_sandbox> set host_type=cipso
tncfg:cipso_sandbox> set doi=1
tncfg:cipso_sandbox> set min_sl="SBX"
tncfg:cipso_sandbox> set max_sl="SBX"
tncfg:cipso_sandbox> exit
```

▼ How to Add Hosts to the System's Known Network

After you add hosts and groups of hosts to a system's `/etc/hosts` file, the hosts are known to the system. Only known hosts can be added to a security template.

Before You Begin You are in the `root` role in the global zone.

1 Add individual hosts to the `/etc/hosts` file.

```
# vi /etc/hosts

...
192.168.111.121   ahost
```

2 Add a group of hosts to the `/etc/hosts` file.

```
# vi /etc/hosts

...
192.168.111.0   111-network
```

Next Steps Continue with "How to Add a Host to a Security Template" on page 207.

▼ How to Add a Host to a Security Template

Before You Begin The following must be in place:

- The security template must exist. For the procedure, see "How to Create Security Templates" on page 203.
- The IP addresses must exist in the `/etc/hosts` file or be resolvable by DNS.

  For the `hosts` file, see "How to Add Hosts to the System's Known Network" on page 206.

  For DNS, see Chapter 3, "Managing DNS (Tasks)," in *Oracle Solaris Administration: Naming and Directory Services*.
- The label endpoints must match. For the rules, see "Overview of Routing in Trusted Extensions" on page 190.
- You must be in the Security Administrator role in the global zone.

1 Add a host name or IP address to a security template.

For example, add the `192.168.1.2` IP address.

```
# tncfg -t cipso
tncfg:cipso> add host=192.168.1.2
```

If you add a host that was previously added to another template, you are notified that you are replacing its security template assignment. For example:

```
# tncfg -t cipso
tncfg:cipso> add host=192.168.1.22
192.168.1.2 previously matched the admin_low template
tncfg:cipso> info
...
host=192.168.1.2/32
tncfg:cipso> exit
```

2 View the changed security template.

For example, the following shows the `192.168.1.2` address added to the `cipso` template:

```
tncfg:cipso> info
...
   host=192.168.1.2/32
```

The prefix length of `/32` indicates that the address is exact.

3 Commit the change and exit the security template.

```
tncfg:cipso> commit
tncfg:cipso> exit
```

To remove a host entry, see Example 16–11.

Example 16–7 Creating a Gateway That Handles Packets at One Label

In Example 16–1, the administrator creates a security template that defines a gateway that can only pass packets at the label `PUBLIC`. In this example, the security administrator ensures that the gateway host's IP address can be resolved.

```
# arp 192.168.131.75
gateway-1.example.com (192.168.131.75) at 0:0:0:1:ab:cd
```

The `arp` command verifies that the host is defined in the system's `/etc/hosts` file or is resolvable by DNS.

Then, the administrator adds the `gateway-1` host to the security template:

```
# tncfg -t cipso_public
tncfg:cipso_public> add host=192.168.131.75
tncfg:cipso_public> exit
```

The system can immediately send and receive `public` packets through `gateway-1`.

Example 16–8 Creating an Unlabeled Router to Route Labeled Packets

In Example 16–2, the administrator creates the security template for the router. In this example, the administrator ensures that the router's IP address can be resolved.

```
# arp 192.168.131.82
router-1.example.com (192.168.131.82) at 0:0:0:2:ab:cd
```

The `arp` command indicates that the host is defined in the system's `/etc/hosts` file or is resolvable by DNS.

Then, the administrator adds the router to the security template.

```
# tncfg -t unl_public
tncfg:unl_public> add host=192.168.131.82
tncfg:unl_public> exit
```

The system can immediately send and receive packets at all labels through `router-1`.

Example 16–9 Creating a Gateway With a Limited Label Range

In Example 16–3, the administrator creates a security template with a limited label range. In this example, the security administrator ensures that the gateway host's IP address can be resolved.

```
# arp 192.168.131.78
gateway-ir.example.com (192.168.131.78) at 0:0:0:3:ab:cd
```

The `arp` command indicates that the host is defined in the system's `/etc/hosts` file or is resolvable by DNS.

Then, the administrator adds the gateway to the security template.

```
# tncfg -t cipso_iuo_rstrct
tncfg:cipso_iuo_rstrct> add host=192.168.131.78
tncfg:cipso_iuo_rstrct> exit
```

The system can immediately send and receive packets that are labeled `internal` and `restricted` through `gateway-ir`.

Example 16–10 Creating a Labeled Host for Developers

In Example 16–6, the administrator creates the `cipso_sandbox` security template. In this example, the security administrator adds two hosts to the template.

```
# tncfg -t cipso_sandbox
tncfg:cipso_sandbox> add host=196.168.129.102
tncfg:cipso_sandbox> add host=196.168.129.129
tncfg:cipso_sandbox> exit
```

The developers who use the `196.168.129.102` and `196.168.129.129` systems can communicate with each other at the label `SANDBOX`.

Example 16–11 Removing Several Hosts From a Security Template

In this example, the security administrator removes several hosts from the `cipso` security template. The administrator uses the `info` subcommand to display the hosts, then types `remove`, and copies and pastes four `host=` entries.

```
# tncfg -t cipso info
   name=cipso
   host_type=cipso
   doi=1
   min_label=ADMIN_LOW
   max_label=ADMIN_HIGH
   host=127.0.0.1/32
   host=192.168.1.2/32
   host=192.168.113.0/24
   host=192.168.113.100/25
   host=2001:a08:3903:200::0/56

# tncfg -t cipso
tncfg:cipso> remove host=192.168.1.2/32
tncfg:cipso> remove host=192.168.113.0/24
tncfg:cipso> remove host=192.168.113.100/25
tncfg:cipso> remove host=2001:a08:3903:200::0/56
tncfg:cipso> info
...
   max_label=ADMIN_HIGH
   host=127.0.0.1/32
   host=192.168.75.0/24
```

After removing the hosts, the administrator commits the changes and exits the security template.

```
tncfg:cipso> commit
tncfg:cipso> exit
#
```

▼ How to Add a Range of Hosts to a Security Template

Before You Begin For the requirements, see "How to Add a Host to a Security Template" on page 207

1 To assign a security template to a subnet, add the subnet address to the template.

For example, add two subnets to the `cipso` template, then display the security template.

```
# tncfg -t cipso
tncfg:cipso> add host=192.168.75.0
tncfg:cipso> add host=192.168.113.0
tncfg:cipso> info
...
host=192.168.75.0/24
host=192.168.113.0/24
tncfg:cipso> exit
```

The prefix length of `/24` indicates that the address, which ends in `.0`, is a subnet.

Note – If you add a range of hosts that was previously added to another template, you are notified that you are replacing its security template assignment.

```
# tncfg -t cipso
tncfg:cipso> add host=192.168.113.100/25
192.168.113.100/25 previously matched the admin_low template
```

2 To assign a security template to a group of contiguous IP addresses, specify the IP address and the prefix length.

In the following example, the prefix length covers the address range of `192.168.113.0` to `192.168.113.127`. The address includes `192.168.113.100`.

```
# tncfg -t cipso
tncfg:cipso> add host=192.168.113.100/25
tncfg:cipso> exit
```

In the following example, the prefix length covers contiguous IPv6 addresses from `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`.

```
# tncfg -t cipso
tncfg:cipso> add host=2001:a08:3903:200::0/56
tncfg:cipso> info
...
host=2001:a08:3903:200::0/56
tncfg:cipso> exit
```

If you mistype an entry, you receive a message similar to the following:

```
# tncfg -t cipso
tncfg:cipso> add host=2001:a08:3903::0/56
Invalid host: 2001:a08:3903::0/56
```

If you add a host that was previously added to another template, you are notified that you are replacing its security template assignment. For example:

```
# tncfg -t cipso
tncfg:cipso> add host=192.168.113.100/32
192.168.113.100/32 previously matched the admin_low template
tncfg:cipso> info
...
host=192.168.113.100/32
tncfg:cipso> exit
```

The Trusted Extensions fallback mechanism ensures that this explicit assignment overrides the previous assignment, as discussed in "Trusted Network Fallback Mechanism" on page 188.

Example 16–12 Creating Hosts at Discrete Labels

In Example 16–4, the administrator creates the security template that recognizes two labels. In this example, the security administrator ensures that each host's IP addresses can be resolved.

```
# arp 192.168.132.21
host-auxset1.example.com (192.168.132.21) at 0:0:0:4:ab:cd
# arp 192.168.132.22
host-auxset2.example.com (192.168.132.22) at 0:0:0:5:ab:cd
# arp 192.168.132.23
host-auxset3.example.com (192.168.132.23) at 0:0:0:6:ab:cd
# arp 192.168.132.24
host-auxset4.example.com (192.168.132.24) at 0:0:0:7:ab:cd
```

The `arp` command indicates that the hosts are defined in the system's `/etc/hosts` file or are resolvable by DNS.

Then, the administrator assigns the range of IP addresses to the security template by using a prefix length.

```
# tncfg -t cipso_int_rstrct
tncfg:cipso_int_rstrct> set host=192.168.132.0/24
```

Example 16–13 Creating an Unlabeled Subnetwork at the Label `PUBLIC`

In Example 16–5, the administrator creates a security template that defines a single-label `PUBLIC` host. In this example, the security administrator assigns a subnetwork to the `PUBLIC` label. Users on the trusted system can mount file systems from this subnetwork at the label `PUBLIC`.

```
# tncfg -t public
tncfg:public> add host=10.10.0.0/16
tncfg:public> exit
```

The subnetwork can immediately be reached at the label PUBLIC.

▼ 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, the admin_low default security template defines every host on the network. Use this procedure to enumerate specific unlabeled hosts.

The local trusted network values on each system are 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 remote host that is not otherwise defined (0.0.0.0/0) 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 0.0.0.0/0 wildcard entry after the system is installed. The entry must be replaced with entries for every host that the system contacts at boot time.

For example, DNS servers, home directory servers, audit servers, broadcast and multicast addresses, and routers must be explicitly added to a template after the 0.0.0.0/0 wildcard entry is removed.

If an application initially recognizes clients at the host address 0.0.0.0/32, then you must add the 0.0.0.0/32 host entry to the admin_low template. 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 /etc/hosts file.

1 Assign the admin_low template to every unlabeled host that must be contacted at boot time.

- Include every unlabeled host that must be contacted at boot time.
- Include every on-link router that is not running Trusted Extensions, through which this system must communicate.
- Remove the 0.0.0.0/0 assignment.

2 Add hosts to the cipso template.

Add each labeled host that must be contacted at boot time.

- Include every on-link router that is running Trusted Extensions, through which this system must communicate.
- Make sure that all network interfaces are assigned to the template.
- Include broadcast addresses.
- Include the ranges of labeled hosts that must be contacted at boot time.

See Example 16–15 for a sample database.

3 Verify that the host assignments allow the system to boot.

Example 16–14 Changing the Label of the `0.0.0.0/0` IP Address

In this example, the administrator creates a public gateway system. The administrator removes the `0.0.0.0/0` host entry from the `admin_low` template and adds the `0.0.0.0/0` host entry to the unlabeled `public` template. The system then recognizes any host that is not specifically assigned to another security template as an unlabeled system with the security attributes of the `public` security template.

```
# tncfg -t admin_low info
tncfg:admin_low> remove host=0.0.0.0      Wildcard address
tncfg:admin_low> exit

# tncfg -t public
tncfg:public> set host_type=unlabeled
tncfg:public> set doi=1
tncfg:public> set def_label="public"
tncfg:public> set min_sl="public"
tncfg:public> set max_sl="public"
tncfg:public> add host=0.0.0.0     Wildcard address
tncfg:public> exit
```

Example 16–15 Enumerating Computers to Contact at Boot Time

In the following example, the administrator configures the trusted network of a Trusted Extensions system with two network interfaces. The system communicates with another network and with routers. The remote hosts are assigned to one of three templates, `cipso`, `admin_low`, or `public`. The following commands are annotated.

```
# tncfg -t cipso
tncfg:admin_low> add host=127.0.0.1     Loopback address
tncfg:admin_low> add host=192.168.112.111      Interface 1 of this host
tncfg:admin_low> add host=192.168.113.111      Interface 2 of this host
tncfg:admin_low> add host=192.168.113.6      File server
tncfg:admin_low> add host=192.168.112.255      Subnet broadcast address
tncfg:admin_low> add host=192.168.113.255      Subnet broadcast address
```

```
tncfg:admin_low> add host=192.168.113.1      Router
tncfg:admin_low> add host=192.168.117.0/24      Another Trusted Extensions network
tncfg:admin_low> exit

# tncfg -t public
tncfg:public> add host=192.168.112.12      Specific network router
tncfg:public> add host=192.168.113.12      Specific network router
tncfg:public> add host=224.0.0.2      Multicast address
tncfg:admin_low> exit

# tncfg -t admin_low
tncfg:admin_low> add host=255.255.255.255      Broadcast address
tncfg:admin_low> exit
```

After specifying the hosts to contact at boot time, the administrator removes the `0.0.0.0/0` entry from the `admin_low` template.

```
# tncfg -t admin_low
tncfg:admin_low> remove host=0.0.0.0
tncfg:admin_low> exit
```

Example 16–16 Making the Host Address `0.0.0.0/32` a Valid Initial Address

In this example, the security administrator configures an application server to accept initial connection requests from potential clients.

The administrator configures the server's trusted network. The server and client entries are annotated.

```
# tncfg -t cipso info
   name=cipso
   host_type=cipso
   doi=1
   min_label=ADMIN_LOW
   max_label=ADMIN_HIGH
   host=127.0.0.1/32
   host=192.168.128.1/32      Application server address
   host=192.168.128.0/24      Application's client network
     Other addresses to be contacted at boot time

# tncfg -t admin_low info
   name=cipso
   host_type=cipso
   doi=1
   def_label=ADMIN_LOW
   min_label=ADMIN_LOW
   max_label=ADMIN_HIGH
   host=192.168.128.0/24      Application's client network
   host=0.0.0.0/0      Wildcard address
     Other addresses to be contacted at boot time
```

After this phase of testing succeeds, the administrator locks down the configuration by removing the default wildcard address, `0.0.0.0/0`, committing the change, and then adding the specific address.

```
# tncfg -t admin_low info
tncfg:admin_low> remove host=0.0.0.0
tncfg:admin_low> commit
tncfg:admin_low> add host=0.0.0.0/32      For initial client contact
tncfg:admin_low> exit
```

The final `admin_low` configuration appears similar to the following:

```
# tncfg -t admin_low
   name=cipso
   host_type=cipso
   doi=1
   def_label=ADMIN_LOW
   min_label=ADMIN_LOW
   max_label=ADMIN_HIGH
   192.168.128.0/24      Application's client network
   host=0.0.0.0/32       For initial client contact
    Other addresses to be contacted at boot time
```

The `0.0.0.0/32` entry allows only the clients of the application to reach the application server.

Configuring Routes and Multilevel Ports (Tasks)

Static routes enable labeled packets to reach their destination through labeled and unlabeled gateways. MLPs enable an application to use one entry point to reach all zones.

▼ How to Add Default Routes

Before You Begin You must be in the Security Administrator role in the global zone.

You have added each destination host, network, and gateway to a security template. For details, see "How to Add a Host to a Security Template" on page 207 and "How to Add a Range of Hosts to a Security Template" on page 210.

1. **Use the `txzonemgr` GUI to create default routes.**

   ```
   # txzonemgr &
   ```

2. **Double-click the zone whose default route you want to set, then double-click its IP address entry.**

   If the zone has more than one IP address, choose the entry with the desired interface.

3. **At the prompt, type the IP address of the router and click OK.**

Note – To remove or modify the default router, remove the entry, create the IP entry again and add the router. If the zone has only one IP address, you must remove the IP instance to remove the entry.

Example 16–17 Using the `route` Command to Set the Default Route for the Global Zone

In this example, the administrator uses the `route` command to create a default route for the global zone.

```
# route add default 192.168.113.1 -static
```

▼ How to Create a Multilevel Port for a Zone

You can add private and shared MLPs to labeled zones and the global 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.

Before You Begin You must be in the `root` role in the global zone. The system must have at least two IP addresses and the labeled zone is halted.

1 Add the proxy host and the web services host to the `/etc/hosts` file.

```
## /etc/hosts file
...
proxy-host-name  IP-address
web-service-host-name IP-address
```

2 Configure the zone.

For example, configure the `public` zone to recognize packets that are explicitly labeled `PUBLIC`. For this configuration, the security template is named `webprox`.

```
# tncfg -t webprox
tncfg:public> set name=webprox
tncfg:public> set host_type=cipso
tncfg:public> set min_label=public
tncfg:public> set max_label=public
tncfg:public> add host=mywebproxy.oracle.com      host name associated with public zone
tncfg:public> add host=10.1.2.3/16      IP address of public zone
tncfg:public> exit
```

3 Configure the MLP.

For example, the web proxy service might communicate with the `PUBLIC` zone over the `8080/tcp` interface.

```
# tncfg -z public add mlp_shared=8080/tcp
# tncfg -z public add mlp_private=8080/tcp
```

4 **To add the MLPs to the kernel, boot the zone.**

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

5 **In the global zone, add routes for the new addresses.**

To add routes, perform "How to Add Default Routes" on page 215.

Example 16–18 Configuring an MLP by Using the `txzonemgr` GUI

The administrator configures the web proxy service by opening the Labeled Zone Manager.

```
# txzonemgr &
```

The administrator double-clicks the `PUBLIC` zone, then double-clicks `Configure Multilevel Ports`. Then the administrator selects and double-clicks the `Private interfaces` line. The selection changes to an entry field similar to the following:

```
Private interfaces:111/tcp;111/udp
```

The administrator starts the web proxy entry with a semicolon separator

```
Private interfaces:111/tcp;111/udp;8080/tcp
```

After completing the private entry, the administrator types the web proxy into the `Shared interfaces` field.

```
Shared interfaces:111/tcp;111/udp;8080/tcp
```

A popup message indicates that the multilevel ports for the `public` zone will be active at the next boot of the zone.

Example 16–19 Configuring a Private Multilevel Port for NFSv3 Over `udp`

In this example, the administrator enables NFSv3 read-down mounts over `udp`. The administrator has the option of using the `tncfg` command.

```
# tncfg -z global add mlp_private=2049/udp
```

The `txzonemgr` GUI provides another way to define the MLP.

In the Labeled Zone Manager, the administrator double-clicks the `global` zone, then double-clicks `Configure Multilevel Ports`. In the MLP menu, the administrator selects and double-clicks the `Private interfaces` line and adds the port/protocol.

```
Private interfaces:111/tcp;111/udp;8080/tcp
```

A popup message indicates that the multilevel ports for the `global` zone will be active at the next boot.

Example 16–20 Displaying Multilevel Ports on a System

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
```

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 219
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 220

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 multilevel 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 are in the `root` role in the global zone.

1 Add the `enigma` and `partym` hosts to a CIPSO security template.

Follow the procedures in "Labeling Hosts and Networks (Task Map)" on page 200. 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 *Oracle Solaris Administration: 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 *Oracle Solaris Administration: 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
    #
## 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 3des }
    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 sha1 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
    #
## Use IKE to exchange security labels.
    label_aware
    #
        p1_xform
          { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
    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 sha1 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 *Oracle Solaris Administration: 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 *Oracle Solaris Administration: IP Services*.

Assume the following modifications to the illustration:

- The `10` subnets are multilevel trusted networks. CIPSO IP option security labels are visible on these LANs.
- The `192.168` subnets are single-label untrusted networks that operate at the `PUBLIC` label. These networks do not support CIPSO IP options.
- Labeled traffic between `euro-vpn` and `calif-vpn` is protected against unauthorized changes.

Before You Begin You are in the `root` role in the global zone.

1 Follow the procedures in "Labeling Hosts and Networks (Task Map)" on page 200 to define the following:

a. Add `10.0.0.0/8` IP addresses to a labeled security template.

Use a template with a CIPSO host type. Retain the default label range, `ADMIN_LOW` to `ADMIN_HIGH`.

b. Add `192.168.0.0/16` IP addresses to an unlabeled security template at label `PUBLIC`.

Use a template with an Unlabeled host type. Set the default label to be `PUBLIC`. Retain the default label range, `ADMIN_LOW` to `ADMIN_HIGH`.

c. Add the `Calif-vpn` and `Euro-vpn` Internet-facing addresses, `192.168.13.213` and `192.168.116.16`, to a CIPSO template.

Retain the default label range.

2 Create an IPsec tunnel.

Follow the procedure in "How to Protect a VPN With IPsec in Tunnel Mode" in *Oracle Solaris Administration: 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 *Oracle Solaris Administration: 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 `euro-vpn` system's `/etc/inet/ike/config` file.

The resulting file appears similar to the following. The label additions are highlighted.

```
        ### ike/config file on euro-vpn, 192.168.116.16
    ## Global parameters
    #
```

```
## 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 3des }
    p2_pfs 2
    #
   ## The rule to communicate with calif-vpn
       # Label must be unique
    { label "eurovpn-califvpn"
          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 sha1 encr_alg aes }
          p2_pfs 5
        }
```

b. Add the same keywords to the `ike/config` file on the `calif-vpn` system.

```
    ### ike/config file on calif-vpn, 192.168.13.213
    ## Global Parameters
    #
## Use IKE to exchange security labels.
    label_aware
    #
        p1_xform
          { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
    p2_pfs 2
    ## The rule to communicate with euro-vpn
    # Label must be unique
    { label "califvpn-eurovpn"
          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 sha1 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 *Oracle Solaris Administration: IP Services*.

Troubleshooting the Trusted Network (Task Map)

The following task map describes tasks to help you debug your Trusted Extensions network.

Task	Description	For Instructions
Determine why a system and a remote host cannot communicate.	Checks that the interfaces on a single system are up.	"How to Verify That a System's Interfaces Are Up" on page 223
	Uses debugging tools when a system and a remote host cannot communicate with each other.	"How to Debug the Trusted Extensions Network" on page 224
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's Connection to the LDAP Server" on page 227

▼ How to Verify That a System'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 attribute values. The Security Administrator role and the System Administrator role can check these values.

1 Verify that the system's network interface is up.

You can use the Labeled Zone Manager GUI or the `ipadm` command to display the system's interfaces.

- **Open the Labeled Zone Manager, then double-click the zone of interest.**

  ```
  # txzonemgr &
  ```

  Select Configure Network Interfaces and verify that the value of the `Status` column for the zone is `Up`.

- **Or, use the `ipadm show-addr` command.**

  ```
  # ipadm show-addr
  ...
  ADDROBJ           TYPE      STATE        ADDR
  lo0/v4            static    ok           127.0.0.1/8
  net0/_a           dhcp      down         10.131.132.133/23
  net0:0/_a         dhcp      down         10.131.132.175/23
  ```

  The value of the `net0` interfaces should be `ok`. For more information about the `ipadm` command, see the `ipadm(1M)` man page.

2 If the interface is not up, bring it up.

a. In the Labeled Zone Manager GUI, double-click the zone whose interface is down.

b. **Select Configure Network Interfaces.**

c. **Double-click the interface whose state is `Down`.**

d. **Select Bring Up, then OK.**

e. **Click Cancel or OK.**

▼ How to Debug the Trusted Extensions Network

To debug two hosts that should be communicating but are not, you can use Trusted Extensions and Oracle Solaris debugging tools. For example, Oracle 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 Appendix D, "List of Trusted Extensions Man Pages."

- For problems with contacting labeled zones, see "Managing Zones (Task Map)" on page 161.
- For debugging NFS mounts, see "How to Troubleshoot Mount Failures in Trusted Extensions" on page 179.

Before You Begin You must be in the global zone in a role that can check network attribute values. The Security Administrator role or the System Administrator role can check these values. Only the `root` role can edit files.

1 **Check that the hosts that cannot communicate are using the same naming service.**

a. **On each system, check the values for the Trusted Extensions databases in the `name-service/switch` SMF service.**

```
# svccfg -s name-service/switch listprop config
config/value_authorization  astring  solaris.smf.value.name-service.switch
config/default              astring  ldap
...
config/tnrhtp               astring  "files ldap"
config/tnrhdb               astring  "files ldap"
```

b. **If the values are different on different hosts, correct the values on the offending hosts.**

```
# svccfg -s name-service/switch setprop config/tnrhtp="files ldap"
# svccfg -s name-service/switch setprop config/tnrhdb="files ldap"
```

c. **Then, restart the naming service daemon on those hosts.**

```
# svcadm restart name-service/switch
```

2 Verify that each host is defined correctly by displaying the security attributes for the source, destination, and gateway hosts in the transmission.

Use the command line to check that the network information is correct. Verify that the assignment on each host matches the assignment on the other hosts on the network. Depending on the view you want, use the `tncfg` command, the `tninfo` command, or the `txzonemgr` GUI.

- **Display a template definition.**

  The `tninfo -t` command displays the labels in string and hexadecimal format.

  ```
  $ 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 a template and the hosts that are assigned to it.**

  The `tncfg -t` command displays the labels in string format and lists the assigned hosts.

  ```
  $ tncfg -t template info
     name=<template-name>
     host_type=<one of cipso or unlabeled>
     doi=1
     min_label=<minimum-label>
     max_label=<maximum-label>
     host=127.0.0.1/32            /** Localhost **/
     host=192.168.1.2/32          /** LDAP server **/
     host=192.168.1.22/32         /** Gateway to LDAP server **/
     host=192.168.113.0/24        /** Additional network **/
     host=192.168.113.100/25            /** Additional network **/
     host=2001:a08:3903:200::0/56      /** Additional network **/
  ```

- **Display the IP address and the assigned security template for a specific host.**

  The `tninfo -h` command displays the IP address of the specified host and the name of its assigned security template.

  ```
  $ tninfo -h hostname
  IP Address: IP-address
  Template: template-name
  ```

  The `tncfg get host=` command displays the name of the security template that defines the specified host.

  ```
  $ tncfg get host=hostname|IP-address[/prefix]
  template-name
  ```

- **Display the multilevel ports (MLP)s for a zone.**

  The `tncfg -z` command lists one MLP per line.

  ```
  $ tncfg -z zone-name info [mlp_private | mlp_shared]
  mlp_private=<port/protocol-that-is-specific-to-this-zone-only>
  ```

`mlp_shared=<`*port/protocol-that-the-zone-shares-with-other-zones*`>`

The `tninfo -m` command lists the private MLPs in one line and the shared MLPs on a second line. The MLPs are separated by semicolons.

```
$ tninfo -m zone-name
private: ports-that-are-specific-to-this-zone-only
shared: ports-that-the-zone-shares-with-other-zones
```

For a GUI display of the MLPs, use the `txzonemgr` command. Double-click the zone, then select Configure Multilevel Ports.

3 Fix any incorrect information.

a. To change or check network security information, use the trusted network administrative commands, `tncfg` and `txzonemgr`. To verify the syntax of the databases, use the `tnchkdb` command.

For example, the following output shows that a template name, `internal_cipso`, is undefined:

```
# tnchkdb
     checking /etc/security/tsol/tnrhtp ...
     checking /etc/security/tsol/tnrhdb ...
tnchkdb: unknown template name: internal_cipso at line 49
tnchkdb: unknown template name: internal_cipso at line 50
tnchkdb: unknown template name: internal_cipso at line 51
     checking /etc/security/tsol/tnzonecfg ...
```

The error indicates that the `tncfg` and `txzonemgr` commands were not used to create and assign the `internal_cipso` security template.

To repair, replace the `tnrhdb` file with the original file, then use the `tncfg` command to create and assign security templates.

b. To clear the kernel cache, reboot.

At boot time, the cache is populated with database information. The SMF service, `name-service/switch`, determines if local or LDAP databases are used to populate the kernel.

4 Collect transmission information to assist in debugging.

a. Verify your routing configuration.

`$ route get [ip] -secattr sl=`*label*`,doi=`*integer*

For details, see the `route(1M)` man page.

b. View the label information in packets.

```
$ snoop -v
```

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.

c. View the routing table entries and the security attributes on sockets.

```
$ netstat -aR
```

The `-aR` option displays extended security attributes for sockets.

```
$ netstat -rR
```

The `-rR` option displays routing table entries. For details, see the `netstat(1M)` man page.

How to Debug a Client's Connection to the LDAP Server

Misconfiguration of a 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.

a. Use the `tncfg` or `tninfo` command to view information.

```
# tncfg get host=LDAP-server
# tncfg get host=gateway-to-LDAP-server

# tninfo -h LDAP-server
# tninfo -h gateway-to-LDAP-server
```

b. Determine the route to the server.

```
# route get LDAP-server
```

If a template assignment is incorrect, add the host to the correct template.

2 Check and if necessary, 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 the `/etc/hosts` file.

3 If you are using DNS, check the configuration of the `svc:/network/dns/client` service.

```
# svccfg -s dns/client listprop config
config                      application
config/value_authorization  astring       solaris.smf.value.name-service.dns.switch
config/nameserver           astring       192.168.8.25 192.168.122.7
```

4 To change the values, use the `svccfg` command.

```
# svccfg -s dns/client setprop config/search = astring: example1.domain.com
# svccfg -s dns/client setprop config/nameserver = net_address: 192.168.8.35
# svccfg -s dns/client:default refresh
# svccfg -s dns/client:default validate
# svcadm enable dns/client
# svcadm refresh name-service/switch
# nslookup some-system
Server:         192.168.135.35
Address:        192.168.135.35#53

Name:   some-system.example1.domain.com
Address: 10.138.8.22
Name:   some-system.example1.domain.com
Address: 10.138.8.23
```

5 Verify that the `tnrhdb` and `tnrhtp` entries in the `name-service/switch` service are accurate.

In the following output, the `tnrhdb` and `tnrhtp` entries are not listed. Therefore, these databases are using the default, `files ldap` naming services, in that order.

```
# svccfg -s name-service/switch listprop config
config                      application
config/value_authorization  astring       solaris.smf.value.name-service.switch
config/default              astring       "files ldap"
config/host                 astring       "files dns"
config/netgroup             astring       ldap
```

6 Check that the client is correctly configured on the server.

```
# ldaplist -l tnrhdb client-IP-address
```

7 Check that the interfaces for your labeled zones are correctly configured on the LDAP server.

```
# ldaplist -l tnrhdb client-zone-IP-address
```

8 Verify that you can contact 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
...
```

9 Configure LDAP and reboot.

a. For the procedure, see "Make the Global Zone an LDAP Client in Trusted Extensions" on page 89.

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 and reboot.

```
# zoneadm list
zone1
zone2
,
,
,
# zoneadm -z zone1 halt
# zoneadm -z zone2 halt
.
.
.
# reboot
```

You could instead use the `txzonemgr` GUI to halt the labeled zones.

◆ ◆ ◆ CHAPTER 17

Trusted Extensions and LDAP (Overview)

This chapter describes the use of the Oracle Directory Server Enterprise Edition (Directory Server) for a system that is configured with 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. The `svc:/system/name-service/switch` service 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 Oracle 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 during system configuration.

Trusted Extensions adds two trusted network databases to the Directory Server: `tnrhdb` and `tnrhtp`.

- For information about the use of the LDAP naming service in Oracle Solaris, see *Oracle Solaris Administration: Naming and Directory Services*.
- Setting up the Directory Server for Trusted Extensions is described in Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)." Trusted Extensions systems can be clients of an Oracle Solaris Directory Server by using a Directory Server proxy that is configured with Trusted Extensions.
- Setting up clients of the Trusted Extensions Directory Server is described in "Creating a Trusted Extensions LDAP Client" on page 89.

Note – Systems that are configured with Trusted Extensions cannot be clients of NIS masters.

Locally Managed Trusted Extensions Systems

If a naming service is not used at a site, administrators must ensure that configuration information for users, systems, and networks is identical on all systems. A change that is made on one system must be made on all systems.

On a locally managed Trusted Extensions system, configuration information is maintained in files in the `/etc`, `/etc/security`, and `/etc/security/tsol` directories.

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 Oracle Solaris. 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 *Oracle Solaris Administration: Naming and Directory Services*.
- To troubleshoot client-to-server LDAP connection problems that are affected by labels, see "How to Debug a Client's Connection to the LDAP Server" on page 227.
- To troubleshoot other client-to-server LDAP connection problems, see Chapter 13, "LDAP Troubleshooting (Reference)," in *Oracle Solaris Administration: Naming and Directory Services*.
- 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 6 or 7 of Oracle Directory Server Enterprise Edition 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 or 7 of Oracle Directory Server Enterprise Edition software, type:

```
# dpadm start /export/home/ds/instances/your-instance
# dpadm stop /export/home/ds/instances/your-instance
```

◆ ◆ ◆ CHAPTER 18

Multilevel Mail in Trusted Extensions (Overview)

This chapter covers security and multilevel mailers on systems that are configured with 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 136.

Trusted Extensions Mail Features

In Trusted Extensions, the System Administrator role sets up and administers mail servers according to instructions in Chapter 13, "Mail Services (Tasks)," in *Oracle Solaris Administration: Network 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(1)` man page. For configuration suggestions, see "`.copy_files` and `.link_files` Files" on page 130.

- 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 Thunderbird mail for multilevel mail requires that you configure a Thunderbird 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.

- 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 "Trusted Extensions Accreditation Checks" on page 191.
  - 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 19

Managing Labeled Printing (Tasks)

This chapter describes how to print labels on Trusted Extensions print output.

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.

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 Oracle Solaris printer administration procedures, then they assign labels to the print servers and printers.

Trusted Extensions software supports both single-level and multilevel printing. By default, single-level printing is configured. Multilevel printing is implemented in the global zone only. To use the global zone's print server, a labeled zone must be configured as an IP instance or as a vnic. The address must 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, an Oracle Solaris print server can be assigned an arbitrary label. Users can then print jobs at that arbitrary label on the Oracle Solaris printer. As with Trusted Extensions printers, those Oracle 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 labels on body pages and banner and trailer pages. The information comes from the `label_encodings` file.

The security administrator can configure user accounts to use printers that do not print labels on the output.

PostScript Printing of Security Information

Labeled printing in Trusted Extensions relies on features from Oracle Solaris printing. In the Oracle Solaris OS, the `job-sheets` option handles banner page creation. To implement labeling, the print job is converted to a PostScript file. Then, the PostScript file is manipulated to insert labels on body pages, and to create banner and trailer pages.

Configuring Labeled Printing (Task Map)

The following task map describes common configuration procedures that are related to labeled printing. For more information, see Chapter 15, "Setting Up and Administering Printers by Using CUPS (Tasks)," in *Oracle Solaris Administration: Common Tasks*.

Note – Printer clients can only print jobs within the label range of the Trusted Extensions print server.

Task	Description	For Instructions
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 240
Configure printing from a labeled zone.	Creates a single–label print server for a labeled zone.	"How to Configure a Zone As a Single-Level Print Server" on page 239
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 241

Task	Description	For Instructions
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 243

▼ How to Configure a Zone As a Single-Level Print Server

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 Your Minimum Label" in *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 *Trusted Extensions User's Guide*.

3 Define the characteristics of every connected printer.

a. At the label of zone, edit the CUPS print server configuration file, `/etc/cups/cupsd.conf`.

4 Assign the appropriate job sheet to each printer that is connected to the print server.

For example, the following specifications create an appropriate labeled sheet:

```
#CUPS-BANNER for INTERNAL print jobs
Show job-id job-name job-originating-user-name job-originating-host-name job-billing
Header CONFIDENTIAL : INTERNAL USE ONLY
Footer CONFIDENTIAL : INTERNAL USE ONLY
Image images/cups.png
```

Use the following command:

```
$ lpadmin -p printer -o job-sheets-default=labeled,labeled
```

The attached printers can print jobs only at the label of the zone.

5 Test the printer.

Note – For security reasons, files with an administrative label, `ADMIN_HIGH` or `ADMIN_LOW`, print `ADMIN_HIGH` on the body of the printout. The banner and trailer pages are labeled with the highest label and compartments in the `label_encodings` file.

As `root` and as a regular user, perform the following steps:

a. Print plain files from the command line.

b. **Print files from your applications, such as Oracle Beehive, your browser, and your editor.**

c. **Verify that labels print correctly.**

See Also

- **Prevent labeled output –** "Reducing Printing Restrictions in Trusted Extensions (Task Map)" on page 244
- **Use this zone as a print server –** "How to Enable a Trusted Extensions Client to Access a Printer" on page 241

▼ 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 **Enable multilevel printing by configuring the global zone with the print server port, `515/tcp`.**

```
# tncfg -z global add mlp_shared=515/tcp
# tncfg -z global add mlp_private=515/tcp
```

2 **Define the characteristics of every connected printer.**

```
# lpadmin -p printer-name -v /dev/null \
-o protocol=tcp -o dest=printer-IP-address:9100 -T PS -I postscript
# accept printer-name
# enable printer-name
```

3 **Configure each printer that is connected to the print server with a labeled job sheet.**

```
$ lpadmin -p printer -o job-sheets-default=labeled,labeled
```

If the default printer label range of `ADMIN_LOW` to `ADMIN_HIGH` is acceptable for every printer, then your label configuration is done.

4 **In every labeled zone where printing is allowed, configure the printer.**

Use the `all-zones` IP address for the global zone as the print server.

a. **Log in as root to the zone console of the labeled zone.**

```
# zlogin -C labeled-zone
```

b. Create an `/etc/cups/client.conf` file in each labeled zone.

This file connects to the `cupsd` daemon in the global zone for print service. Modify this file to include the print server name and its IP address. For information about the configuration file, see the `client.conf`(5) man page.

c. (Optional) Set the printer as the default.

```
# lpadmin -d printer-name
```

5 In every labeled zone, test the printer.

As `root` and as a regular user, perform the following steps:

a. Print plain files from the command line.

b. Print files from your applications, such as Oracle Beehive, your browser, and your editor.

c. Verify that labels print correctly.

See Also

- **Limit printer label range –** "How to Configure a Restricted Label Range for a Printer" on page 243
- **Prevent labeled output –** "Reducing Printing Restrictions in Trusted Extensions (Task Map)" on page 244
- **Use this zone as a print server –** "How to Enable a Trusted Extensions Client to Access a Printer" on page 241

▼ 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 Zone As a Single-Level Print Server" on page 239
- "How to Configure a Multilevel Print Server and Its Printers" on page 240
- "How to Assign a Label to an Unlabeled Print Server" on page 245

You must be in the System Administrator role in the global zone.

1 Complete the procedures that enable your systems to access a printer.

- **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 *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 *Trusted Extensions User's Guide*.

c. Add access to the printer that is connected to the arbitrarily labeled print server.

```
$ lpadmin -s printer
```

2 Test the printers.

Note – For security reasons, files with an administrative label, `ADMIN_HIGH` or `ADMIN_LOW`, print `ADMIN_HIGH` on the body of the printout. The banner and trailer pages are labeled with the highest label and compartments in the `label_encodings` file.

On every client, test that printing works for `root` and roles in the global zone and for `root`, roles, and regular users in labeled zones.

a. Print plain files from the command line.

b. Print files from your applications, such as Oracle Beehive, your browser, and your editor.

c. Verify that labels print correctly.

▼ 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 107.

   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 245
Configure printers at a single label without labeled output.	Enables users to print at a specific label to an Oracle Solaris printer. The print jobs are not marked with labels.	"How to Assign a Label to an Unlabeled Print Server" on page 245
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 246
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 247
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 246
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 247

Task	Description	For Instructions
Assign printing authorizations.	Enables users to bypass default printing restrictions.	"How to Create a Rights Profile for Convenient Authorizations" on page 140 "How to Modify `policy.conf` Defaults" on page 134

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 an Oracle 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

An Oracle 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.

- **Assign an unlabeled template to the print server.**

  For details, see "How to Add a Host to a Security Template" on page 207.

Choose a label. Users who are working at that label can send print jobs to the Oracle 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 19–1 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 Oracle Solaris print server. The template is described in Example 16–5. 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.**

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 140.
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 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 140.
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.**
    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.**

  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 `solaris.print.ps` authorization to those users and roles. For details, see "How to Create a Rights Profile for Convenient Authorizations" on page 140.

Example 19–2 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 20

Devices in Trusted Extensions (Overview)

This chapter describes the extensions that Trusted Extensions provides to device protection.

Device Protection With Trusted Extensions Software

On an Oracle 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 the Trusted Extensions feature uses the device protection mechanisms of the Oracle 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 Manager.

For information about device protection in Oracle Solaris, see Chapter 5, "Controlling Access to Devices (Tasks)," in *Oracle Solaris Administration: 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 Oracle Solaris. 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 *Oracle Solaris Administration: 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. Oracle Solaris 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 *Oracle Solaris Administration: Security Services*.

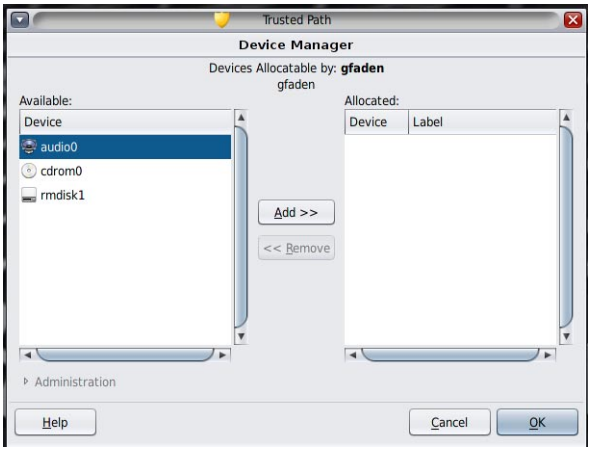
For Trusted Extensions software, device-clean scripts must satisfy certain requirements. These requirements are described in the `device_clean`(5) man page.

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.

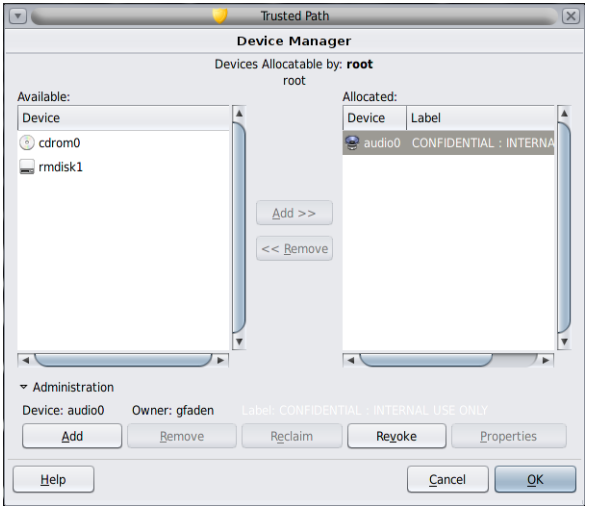
The GUI is called the Device Manager. This GUI is started from the Trusted Path menu by selecting Allocate Device. The following figure shows a Device Manager that was opened by a user who can allocate the `audio` device.

FIGURE 20–1 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 Oracle Solaris interfaces and Trusted Extensions interfaces.

For Oracle Solaris command-line interfaces, see "Device Protection (Reference)" in *Oracle Solaris Administration: 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 *Oracle Solaris Administration: Security Services* and "Forcibly Deallocating a Device" in *Oracle Solaris Administration: Security Services*.

For Trusted Extensions command-line interfaces, see the `add_allocatable(1M)` and `remove_allocatable(1M)` man pages.

◆ ◆ ◆ CHAPTER 21

Managing Devices for Trusted Extensions (Tasks)

This chapter describes how to administer and use devices on a system that is configured with Trusted Extensions.

Handling Devices in Trusted Extensions (Task Map)

The following task map links 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 256
Administer devices.	Configures devices for regular users.	"Managing Devices in Trusted Extensions (Task Map)" on page 256
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 264

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 Oracle Solaris `allocate` command does not work in Trusted Extensions. The following task map links to user procedures for using devices in Trusted Extensions.

Task	For Instructions
Allocate and deallocate a device.	"How to Allocate a Device in Trusted Extensions" in *Trusted Extensions User's Guide*
Use portable media to transfer files.	"How to Copy Files From Portable Media in Trusted Extensions" on page 77 "How to Copy Files to Portable Media in Trusted Extensions" on page 75

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 *Oracle Solaris Administration: Security Services*
Authorize users to allocate a device.	The Security Administrator role assigns a rights profile with the Allocate Device authorization to the user.	"How to Authorize Users to Allocate a Device" in *Oracle Solaris Administration: 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 264
Configure a device.	Chooses security features to protect the device.	"How to Configure a Device in Trusted Extensions" on page 257
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 261
	Uses Oracle Solaris commands to make a device available or unavailable for use.	"Forcibly Allocating a Device" in *Oracle Solaris Administration: Security Services* "Forcibly Deallocating a Device" in *Oracle Solaris Administration: Security Services*
Prevent access to an allocatable device.	Provides fine–grained access control to a device.	Example 21–2
	Denies everyone access to an allocatable device.	Example 21–1

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 262
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 263

▼ 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.

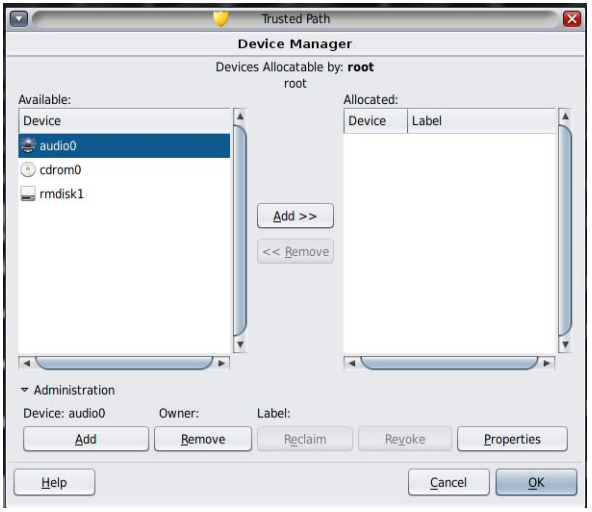
The following devices can be allocated for use:

- `audio`*n* – Indicates a microphone and speaker
- `cdrom`*n* – Indicates a CD-ROM drive
- `floppy`*n* – Indicates a diskette drive
- `mag_tape`*n* – Indicates a tape drive (streaming)
- `rmdisk`*n* – Indicates a removable disk, such as a JAZ or ZIP drive, or USB hot-pluggable media

Before You Begin You must be in the Security Administrator role in the global zone.

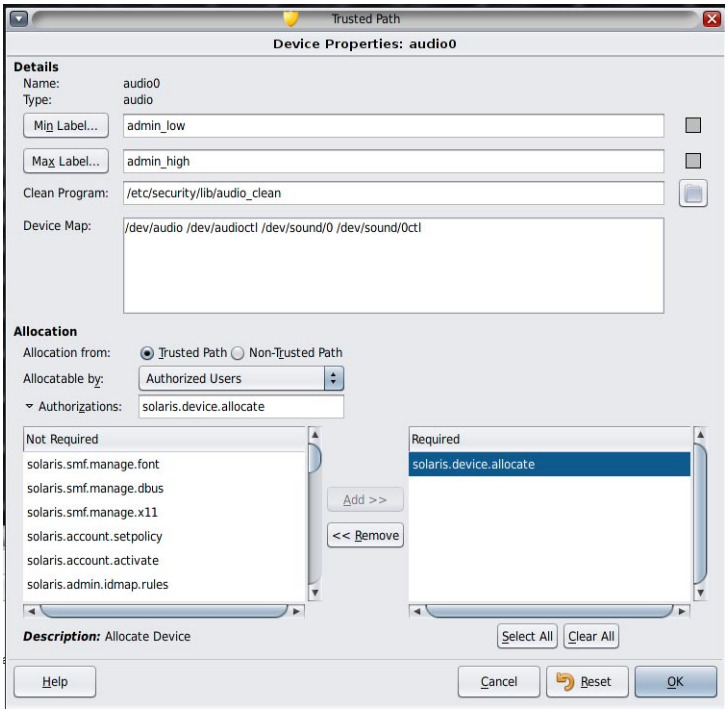
1 **From the Trusted Path menu, select Allocate Device.**

The Device Manager appears.



2 View the default security settings.

Click Administration, then highlight the device. The following figure shows an audio device that is being viewed by the `root` role.



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 107.

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.**

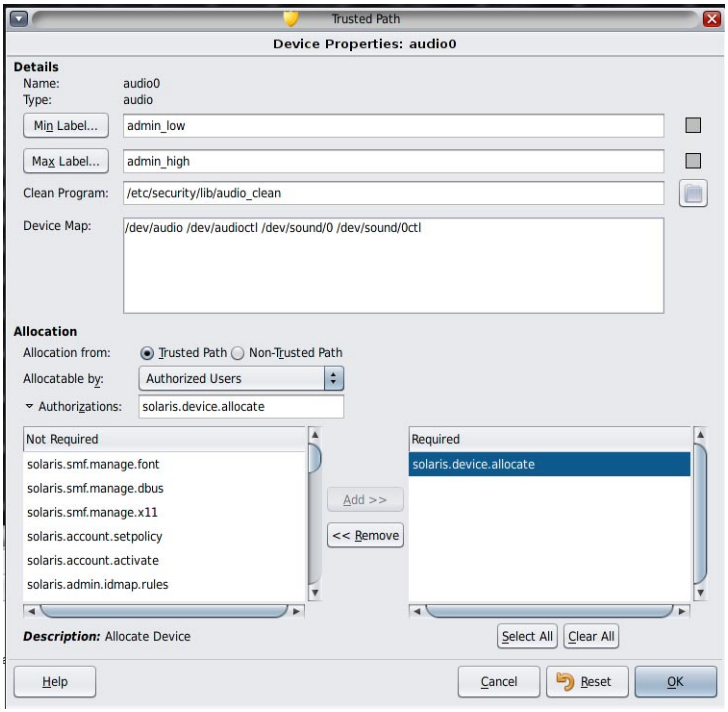
5 **Specify if the device can be allocated remotely.**

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 264.

7 **To save your changes, click OK.**

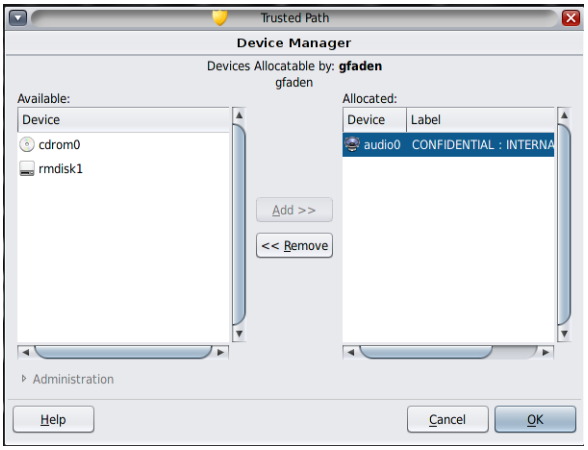
▼ How to Revoke or Reclaim a Device in Trusted Extensions

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 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 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 107.
      ii. **Set the maximum label.**
         Click the Max Label... button. Choose a maximum label from the label builder.

Example 21–1 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 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 `root` role in the global zone.

1. **Copy the script into the `/etc/security/lib` directory.**
2. **In the Device Properties 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 264
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 267
Assign device authorizations to users and roles.	Enables users and roles to use the new authorizations.	"How to Assign Device Authorizations" on page 268

▼ 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 21–1.

Before You Begin You must be in the Security Administrator role in the global zone.

1. **Edit the `auth_attr` file.**

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 Oracle Directory Server Enterprise Edition (Directory 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.**

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 267.

Example 21–2 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 21–3 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 264.

1 Follow the "How to Configure a Device in Trusted Extensions" on page 257 procedure.

a. Select a device that needs to be protected with your new authorizations.

b. Click the Administration button.

c. 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 140.

- **Assign to the user a rights profile that contains the Allocate Device authorization.**

  For the step-by-step procedure, see "How to Change the RBAC Properties of a User" in *Oracle Solaris Administration: Security Services*.

  The following rights profiles enable a role to allocate devices:

  - All Authorizations
  - Device Management
  - Media Backup
  - Media Restore
  - Object Label Management
  - Software Installation

  The following rights profiles enable a role to revoke or reclaim devices:

  - All Authorizations
  - Device Management

  The following rights profiles enable a role to create or configure devices:

  - All Authorizations
  - Device Security

Example 21–4 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 264
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 22

Trusted Extensions Auditing (Overview)

This chapter describes the additions to auditing that 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 an Oracle 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.
- Per-zone auditing is discouraged, because it requires a `root` account in a labeled zone.
- 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. The system administrator plans disk space requirements for the audit files, creates an audit administration server, and reviews audit logs.

Audit Management by Role in Trusted Extensions

Auditing in Trusted Extensions requires the same planning as in the Oracle Solaris OS. For details about planning, see Chapter 27, "Planning for Auditing," in *Oracle Solaris Administration: Security Services*.

Role Responsibilities for Audit Administration

In Trusted Extensions, auditing is the responsibility of separate roles.

- The `root` role assigns audit flags to users and rights profiles, and edits system files, such as the `audit_warn` script.
- The System Administrator role sets up the disks and the network of audit storage. This role can also review the audit records.
- The Security Administrator role decides what is to be audited and configures auditing. The initial setup team created this role by completing"How to Create the Security Administrator Role in Trusted Extensions" on page 66.

Note – A system only records the events in audit classes that the security administrator has preselected. 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

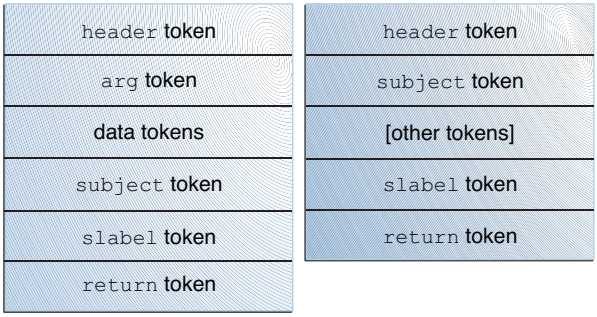
The procedures to configure and manage auditing in Trusted Extensions differ only slightly from Oracle Solaris procedures. In Trusted Extensions, audit configuration is performed in the global zone. Because per-zone auditing is not configured, user actions are audited identically in the global zone and in labeled zones. The label of every audited event is included in the audit record.

- The security administrator can select audit policies that are specific to Trusted Extensions, `windata_down` and `windata_up`.
- When reviewing audit records, the system administrator can select audit records by label. For more information, see the `auditreduce(1M)` man page.

Trusted Extensions Audit Reference

Trusted Extensions software adds audit classes, audit events, audit tokens, and audit policy options to Oracle Solaris. Several auditing commands are extended to handle labels. The following figure shows a typical Trusted Extensions kernel audit record and user-level audit record.

FIGURE 22–1 Typical Audit Record Structures on a Labeled System



Trusted Extensions Audit Classes

Trusted Extensions adds X windows audit classes to Oracle Solaris. The classes are listed in the `/etc/security/audit_class` file. For more information about audit classes, see the `audit_class(4)` man page.

The X server audit events are mapped to these classes according to the following criteria:

- **xa** – This class audits access to the X server, that is, X client connection and X client disconnection.
- **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 Oracle Solaris are listed alphabetically in the following table. The token definitions are listed in the `audit.log(4)` man page.

TABLE 22–1 Trusted Extensions Audit Tokens

Token Name	Description
"`label` Token" on page 274	Sensitivity label
"`xatom` Token" on page 274	X window atom identification
"`xcolormap` Token" on page 274	X window color information
"`xcursor` Token" on page 275	X window cursor information
"`xfont` Token" on page 275	X window font information
"`xgc` Token" on page 275	X window graphical context information
"`xpixmap` Token" on page 275	X window pixel mapping information
"`xproperty` Token" on page 275	X window property information
"`xselect` Token" on page 276	X window data information
"`xwindow` Token" on page 276	X window's window information

`label` Token

The `label` token contains a sensitivity label.

A `label` token is displayed by the `praudit -x` command as follows:

```
<sensitivity_label>ADMIN_LOW</sensitivity_label>
```

`xatom` Token

The `xatom` token identifies an X atom.

An `xatom` token is displayed by `praudit` as follows:

```
X atom,_DT_SAVE_MODE
```

`xcolormap` Token

The `xcolormap` token contains information about the use of colormaps, including the X server identifier and the creator's user ID.

An `xcolormap` token is displayed by `praudit` as follows:

```
<X_colormap xid="0x08c00005" xcreator-uid="srv"/>
```

`xcursor` Token

The `xcursor` token contains information about cursor use, including the X server identifier and 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 font use, including the X server identifier and the creator's user ID.

An `xfont` token is displayed by `praudit` as follows:

```
<X_font xid="0x08c00001" xcreator-uid="srv"/>
```

`xgc` Token

The `xgc` token contains information about the graphic context of an X window.

An `xgc` token is displayed by `praudit` as follows:

```
Xgraphic context,0x002f2ca0,srv

<X_graphic_context xid="0x30002804" xcreator-uid="srv"/>
```

`xpixmap` Token

The `xpixmap` token contains information about the use of pixel mappings, including the X server identifier and the creator's user ID.

An `xpixmap` token is displayed by `praudit -x` as follows:

```
<X_pixmap xid="0x2f002004" xcreator-uid="srv"/>
```

`xproperty` Token

The `xproperty` token contains information about various properties of a window, such as the X server identifier, the creator's user ID, and an atom identifier.

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.

An `xselect` token is displayed by `praudit` as follows:

```
X selection,entryfield,halogen
```

`xwindow` Token

The `xwindow` token identifies the Xserver and the creator's user ID.

An `xwindow` token is displayed by `praudit` as follows:

```
<X_window xid="0x07400001" xcreator-uid="srv"/>
```

Trusted Extensions Audit Policy Options

Trusted Extensions adds two window audit policy options to existing audit policy options.

```
$ 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.

◆ ◆ ◆ CHAPTER 23

Software Management in Trusted Extensions (Reference)

This chapter contains information about ensuring that third-party software runs in a trustworthy manner on a system that is configured with Trusted Extensions.

Adding Software to Trusted Extensions

Any software that can be added to an Oracle 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 an Oracle Solaris system that is running non-global zones.

In Trusted Extensions, programs are typically installed in the global zone for use by regular users in labeled zones. For details about packages and zones, see Chapter 24, "About Automatic Installation and Packages on an Oracle Solaris 11 System With Zones Installed," in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

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.

Security Mechanisms for Oracle Solaris Software

Trusted Extensions uses the same security mechanisms as Oracle Solaris. The mechanisms include the following:

- **Authorizations –** Users of a program can be required to have a particular authorization. For information about authorizations, see "RBAC Elements and Basic Concepts" in *Oracle Solaris Administration: Security Services*. Also, see the `auth_attr(4)` man page.
- **Privileges –** Programs and processes can be assigned privileges. For information about privileges, see Chapter 8, "Using Roles and Privileges (Overview)," in *Oracle Solaris Administration: 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 *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.
- **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 *Oracle Solaris Administration: 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 Oracle Solaris, 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 *Developer's Guide to Oracle Solaris 11 Security*. For examples of code for Trusted Extensions, see *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.

◆ ◆ ◆ 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 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 *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

Sun Microsystems Security Engineers. *Solaris 10 Security Essentials*. Prentice Hall, 2009.

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 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 282.
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.
3. Enable Trusted Extensions. See "Enabling the Trusted Extensions Service and Logging In" on page 47.
   a. Install the Oracle Solaris OS.
   b. Load the Trusted Extensions packages.
   c. Enable `svc:/system/labeld`, the Trusted Extensions service.
   d. Reboot.
4. (Optional) Customize the global zone. See "Setting Up the Global Zone in Trusted Extensions" on page 51.

a. If using IPv6, enable IPv6 for Trusted Extensions.
b. If using a DOI different from 1, set the DOI in the `/etc/system` file and in every security template.
c. Verify and install your site's `label_encodings` file.
d. Reboot.

5. Add labeled zones. See "Creating Labeled Zones" on page 55.
   a. Configure two labeled zones automatically.
   b. Configure your labeled zones manually.
   c. Create labeled workspace.
6. Configure the LDAP naming service. See Chapter 5, "Configuring LDAP for Trusted Extensions (Tasks)."

   Create either a Trusted Extensions proxy server or a Trusted Extensions LDAP server. The files naming service requires no configuration.
7. Configure interfaces and routing for the global zone and for labeled zones. See "Configuring the Network Interfaces in Trusted Extensions" on page 60.
8. Configure the network. See "Labeling Hosts and Networks (Task Map)" on page 200.
   - Identify single-label hosts and limited-range hosts.
   - Determine the labels to apply to incoming data from unlabeled hosts.
   - Customize the security templates.
   - Assign individual hosts to security templates.
   - Assign subnets to security templates.
9. Perform further configurations.
   a. Configure network connections for LDAP.
      - Assign the LDAP server or proxy server to the `cipso` host type in all security templates.
      - Assign LDAP clients to the `cipso` host type in all security templates.
      - Make the local system a client of the LDAP server.
   b. Configure local users and local administrative roles. See "Creating Roles and Users in Trusted Extensions" on page 66.
      - 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.
   c. Create home directories at every label that the user can access. See "Creating Centralized Home Directories in Trusted Extensions" on page 71.
      - Create home directories on an NFS server.
      - Create local ZFS home directories that can be encrypted.
      - (Optional) Prevent users from reading their lower-level home directories.
   d. Configure printing. See "Configuring Labeled Printing (Task Map)" on page 238.

e. Configure devices. See "Handling Devices in Trusted Extensions (Task Map)" on page 255.

   i. Assign the Device Management profile or the System Administrator profile to a role.

   ii. To make devices usable, do one of the following:

   - Per system, make devices allocatable.
   - Assign the Allocate Device authorization to selected users and roles.

f. Configure Oracle Solaris features.

   - Configure auditing.
   - Configure system security values.
   - Enable particular LDAP clients to administer LDAP.
   - Configure users in LDAP.
   - Configure network roles in LDAP.

g. Mount and share file systems. See Chapter 14, "Managing and Mounting Files in Trusted Extensions (Tasks)."

◆ ◆ ◆ APPENDIX C

Quick Reference to Trusted Extensions Administration

Trusted Extensions interfaces extend the Oracle 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, naming services options, and for making the global zone a client of an existing LDAP server. In the Oracle Solaris 11 release, the `txzonemgr -c` command bypasses the menus to create the first two labeled zones.
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. The `tgnome-selectlabel` utility allows you to customize a label builder. See "tgnome-selectlabel Utility" in *Trusted Extensions Developer's Guide*,

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 (Login/Workspace) 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 108.

Oracle Solaris Interfaces Extended by Trusted Extensions

Trusted Extensions adds to existing Oracle Solaris configuration files, commands, and GUIs.

Administrative commands	Trusted Extensions adds options to selected Oracle Solaris commands. For a list of all Trusted Extensions interfaces, see Appendix D, "List of Trusted Extensions Man Pages."
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 File Systems in Trusted Extensions" on page 171. 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 128. Trusted Extensions adds audit tokens, audit events, audit classes, and audit policy options. For a list, see "Trusted Extensions Audit Reference" on page 272.
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 Oracle Solaris OS:

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.
Roles	Roles are available in the Oracle Solaris OS, but their use is optional. In Trusted Extensions, roles are required for proper administration.

Limited Options in Trusted Extensions

Trusted Extensions narrows the range of Oracle Solaris configuration options:

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 Oracle Solaris users are not available to regular Trusted Extensions users. Non-global zones are labeled zones. Users work in labeled zones.

◆ ◆ ◆ APPENDIX D

List of Trusted Extensions Man Pages

Trusted Extensions is a configuration of the Oracle Solaris OS. This appendix provides a description of the man pages that include Trusted Extensions information.

Trusted Extensions Man Pages in Alphabetical Order

The following man pages are relevant only on a system that is configured with Trusted Extensions. The description includes links to examples or explanations of these features in the Trusted Extensions document set.

Trusted Extensions Man Page	Purpose and Links to Additional Information
`add_allocatable(1M)`	Enables a device to be allocated by adding the device to device allocation databases. By default, removable devices are allocatable. See "How to Configure a Device in Trusted Extensions" on page 257.
`atohexlabel(1M)`	Converts a human-readable label to its internal text equivalent. For an example, see "How to Obtain the Hexadecimal Equivalent for a Label" on page 121.
`blcompare(3TSOL)`	Compares binary labels.
`blminmax(3TSOL)`	Determines the bound of two labels.
`chk_encodings(1M)`	Checks the label encodings file syntax.

	For examples, see "How to Debug a label_encodings File" in *Trusted Extensions Label Administration* and Example 4–1.
`fgetlabel`(2)	Gets the file's label
`getlabel`(1)	Displays the label of the selected files or directories. For an example, see "How to Display the Labels of Mounted Files" on page 163.
`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. "Acquiring a Sensitivity Label" in *Trusted Extensions Developer's Guide*
`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 For an example, see "How to Obtain a Readable Label From Its Hexadecimal Form" on page 123.
`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 Trusted Extensions label attributes
`libtsnet`(3LIB)	Is the Trusted Extensions network library

`libtsol`(3LIB)	Is the 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 For an example of its use, see "How to Log In and Administer a Remote Trusted Extensions System" on page 153.
`plabel`(1)	Gets the label of a process
`remove_allocatable`(1M)	Prevents allocation of a device by removing its entry from device allocation databases For an example, see "How to Configure a Device in Trusted Extensions" on page 257.
`sel_config`(4)	Is the selection rules for copy, cut, paste, and drag-and-drop operations See "Rules When Changing the Level of Security for Data" on page 114.
`setflabel`(3TSOL)	Moves a file to a zone with the corresponding sensitivity label
`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.
`str_to_label`(3TSOL)	Parses human-readable strings to a label
`tncfg`(1M)	Manages the trusted network databases. An alternative to the `txzonmgr` GUI for managing the trusted network. The `list` subcommand displays the security characteristics of network interfaces. `tncfg` provides more complete information than the `tninfo` command. For many examples, see Chapter 16, "Managing Networks in Trusted Extensions (Tasks)."
`tnctl`(1M)	Configures Trusted Extensions network parameters. You can also use the `tncfg` command. For an example, see Example 12–1.

`tnd`(1M)	Executes the trusted network daemon when the LDAP naming service is enabled.
`tninfo`(1M)	Displays kernel-level Trusted Extensions network information and statistics. "How to Debug the Trusted Extensions Network" on page 224. You can also use the `tncfg` command and the `txzonemgr` GUI. For a comparison with the `tncfg` command, see "How to Troubleshoot Mount Failures in Trusted Extensions" on page 179.
`trusted_extensions`(5)	Introduces Trusted Extensions
`txzonemgr`(1M)	Manages labeled zones and network interfaces. Command-line options enable automatic creation of two zones. This command accepts a configuration file as input and enables the deletion of zones. `txzonemgr` is a `zenity` (1) script. See "Creating Labeled Zones" on page 55 and "Troubleshooting the Trusted Network (Task Map)" on page 223.
`TrustedExtensionsPolicy`(4)	Is the configuration file for Trusted Extensions X Server Extension
`tsol_getrhtype`(3TSOL)	Gets the host type from Trusted Extensions network information
`tgnome-selectlabel` utility	Enables you to create a label builder GUI For more information, see "tgnome-selectlabel Utility" in *Trusted Extensions Developer's Guide*.
`updatehome`(1)	Updates the home directory copy and link files for the current label See "How to Configure Startup Files for Users in Trusted Extensions" on page 136.
`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

Oracle Solaris Man Pages That Are Modified by Trusted Extensions

Trusted Extensions adds information to the following Oracle Solaris man pages.

Oracle Solaris Man Page	Trusted Extensions Modification and Links to Additional Information
allocate(1)	Adds options to support allocating a device in a zone and cleaning the device in a windowed environment. In Trusted Extensions, regular users do not use this command.

	For the user procedure, see "How to Allocate a Device in Trusted Extensions" in *Trusted Extensions User's Guide*.
`auditconfig`(1M)	Adds the window policy, audit classes, audit events, and audit tokens for labeled information.
`auditreduce`(1M)	Adds the `-l` option to select audit records by label. For examples, see "How to Select Audit Events From the Audit Trail" in *Oracle Solaris Administration: Security Services*.
`auth_attr`(4)	Adds label authorizations
`automount`(1M)	Adds the capability to mount, and therefore view, lower-level home directories. Modifies the names and contents of `auto_home` maps to account for zone names and zone visibility from higher labels. For more information, see "Changes to the Automounter in Trusted Extensions" on page 172.
`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. In Trusted Extensions, regular users do not use this command. For the user procedure, see "How to Allocate a Device in Trusted Extensions" in *Trusted Extensions User's Guide*.
`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
`ikeadm`(1M)	Adds a debug flag, `0x0400`, 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. Also, see the `ike.config`(4) man page.

ipadm(1M)	Adds the `all-zones` interface as a permanent property value. For an example, see "How to Verify That a System's Interfaces Are Up" on page 223.
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.
is_system_labeled(3C)	Determines whether the system is configured with Trusted Extensions
ldaplist(1)	Adds Trusted Extensions network databases in LDAP
list_devices(1)	Adds attributes, such as labels, that are associated with a device. 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.
netstat(1M)	Adds the `-R` option to display extended security attributes for sockets and routing table entries.. For an example, see "How to Troubleshoot Mount Failures in Trusted Extensions" on page 179.
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. Adds the `-secattr` option to display the security attributes of the route: `cipso`, `doi`, `max_sl`, and `min_sl`. For an example, see "How to Troubleshoot Mount Failures in Trusted Extensions" on page 179.
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
socket.h(3HEAD)	Supports the `SO_MAC_EXEMPT` option for unlabeled peers
tar(1)	Adds the `-T` option to archive and extract files and directories that are labeled.

	See "How to Back Up Files in Trusted Extensions" on page 175 and "How to Restore Files in Trusted Extensions" on page 175.
`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 the `idletime`, `idlecmd`, `clearance`, and `min_label` user security attributes that are specific to Trusted Extensions See "Planning User Security in Trusted Extensions" on page 33.

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 Oracle 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.

branded zone In Trusted Extensions, a labeled non-global zone. More generally, a non-global zone that contains non-native operating environments. See the `brands(5)` man page.

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. 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. 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 an Oracle 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.
- Through an Assurance Continuity, the NSA certified Trusted Extensions software that is configured on the Solaris 10 5/09 release.

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 Oracle-specific `LOCAL DEFINITIONS` section to the end of the GFI. For details, see Chapter 5, "Customizing the LOCAL DEFINITIONS Section (Tasks)," in *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 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 an Oracle 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 an Oracle 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 an Oracle 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 an Oracle 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 an Oracle 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. 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 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.

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 security attributes, such as privileges, authorizations, and special UIDs and GIDs. A profile shell typically limits users to fewer commands, but can allow these commands to run with more rights. 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 Oracle 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.

security template A record in the `tnrhtp` database that defines the security attributes of a class of hosts that can access the Trusted Extensions network.

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.

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 as a file in `/etc/security/tsol/tnrhdb`.

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`.

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 an Oracle 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. `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 Oracle Solaris OS.

unlabeled system To an Oracle 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

B

C

D

E

I

K

L

M

N

O

P

R

S

T

U

V

W

X

Z