Solaris Trusted Extensions Transition Guide

Part No: 819–0871–11
June 2007

ORACLE

110417@25097

Contents

Tables

Preface

The *Solaris Trusted Extensions Transition Guide* describes the differences between the Solaris Trusted Extensions release and the following releases:

- The Trusted Solaris 8 releases
- The Solaris 10 Operating System release

  The differences include labels, a modified desktop, a modified windowing system, and modified administration tools.

The most recent changes to the software are described in "Differences Between Trusted Solaris 8 Software and Solaris Trusted Extensions" on page 13.

Note – This Solaris release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC, SPARC64, AMD64, Pentium, and Xeon EM64T. The supported systems appear in the *Solaris 10 Hardware Compatibility List* at `http://www.sun.com/bigadmin/hcl`. This document cites any implementation differences between the platform types.

In this document these x86 related terms mean the following:

- "x86" refers to the larger family of 64-bit and 32-bit x86 compatible products.
- "x64" points out specific 64-bit information about AMD64 or EM64T systems.
- "32-bit x86" points out specific 32-bit information about x86 based systems.

For supported systems, see the *Solaris 10 Hardware Compatibility List*.

Who Should Use This Book

All users should find the book useful. The *Solaris Trusted Extensions Transition Guide* is designed for users who are familiar with Trusted Solaris releases and with the Solaris OS. This book enables these users to more easily use systems that are configured with Solaris Trusted Extensions.

Related Books

If you have used the Trusted Solaris 7 release, but you have not used the Trusted Solaris 8 releases, read the *Trusted Solaris 8 HW 7/03 Transition Guide*. The guide is available on the Sun documentation site.

Documentation, Support, and Training

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

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

Typographic Conventions

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

TABLE P–1 Typographic Conventions

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

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

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

Transition to the Solaris Trusted Extensions Release

This book covers the following topics:

The Solaris Trusted Extensions Release

Solaris Trusted Extensions software is a specific configuration of the Solaris Operating System (Solaris OS). Solaris Trusted Extensions (Trusted Extensions) provides labels for local objects and processes, for the desktop and windowing system, for zones and file systems, and for network communications. Trusted Extensions software is delivered in packages that are added to a version of the Solaris OS.

Trusted Extensions depends on features in the Solaris release to which the Trusted Extensions packages are added. Trusted Extensions software does not replace any Solaris components, but the software does modify certain policy settings.

Overview of Changes From Trusted Solaris Software

Trusted Extensions administrators assign labels to hosts, zones, devices, and users. Trusted Extensions applies these labels to resources such as files, processes, network packets, and windows. The basis for applying these labels is the host or zone with which the resources are associated.

As in previous Trusted Solaris releases, the Solaris OS provides support for privileges, authorizations, and auditing. Trusted Extensions adds to the privileges, authorizations, rights

profiles, audit classes, and audit events that the Solaris OS defines. As in previous releases, Trusted Extensions adds CDE actions to rights profiles.

As in previous releases, the software provides a trusted windowing system, desktop, and administration tools that extend Solaris functionality. Printing is modified to handle labeled print jobs. Also, Trusted Extensions provides a trusted version of the Sun Java Desktop System. This trusted version is called Solaris Trusted Extensions (JDS).

Unlike Trusted Solaris software, Trusted Extensions is a configuration of the underlying Solaris OS. Trusted Extensions does not support the NIS+ naming service. LDAP is the recommended naming service for this release. Also, the root user in Trusted Extensions is identical to the root user in the Solaris OS. You can modify the root user as you can in the Solaris OS, that is, by turning the root user into a role.

Summary of Removed Trusted Solaris Features

Because of changes to the architecture, the following Trusted Solaris 8 features do not exist in Trusted Extensions. For a list of interface changes, see Interface Changes in the Solaris Trusted Extensions Release.

- Forced and allowed privileges
- Selected GUIs
  - Dialog boxes for setting privileges and labels in the File Manager
  - Enable Logins GUI
  - Password generator GUI
- Kernel switch settings in `/etc/system` file

```
tsol_hide_upgraded_names                tsol_clean_windows
tsol_privs_debug                        tsol_flush_buffers
```

- Label and privilege attributes in ufs and tmpfs inodes
- Mail delivery exceptions in `sendmail.cf` and the Rights tool

```
#0 LabelTooLow=return                   tsoltoolowreturn
#0 LabelAdminLow=upgrade                tsoladminlowupgrade
```

- Network labeling protocols – `tsol`, `tsix`, and `ripso` labels
- Multilevel directory adornments, including MLDs and SLDs
- NIS+ naming service for a Trusted Extensions domain
- Packaging utilities, such as `tsolinfo` file
- Trusted Solaris extensions to file system commands (Trusted NFS)
- System_Admin actions for NIS+

- `tnidb` database
- t6 API
- `vfstab_adjunct` file

Because of changes to the architecture, the following Trusted Solaris 8 features are visibly different in Trusted Extensions.

- Privileges are called by name, not by number
- Window label `Trusted Path` replaces `ADMIN_LOW` and `ADMIN_HIGH`
- `Trusted Path` workspaces are used for `ADMIN_LOW` and `ADMIN_HIGH` tasks
- Label attributes are not placed in `exec_attr`
- `tsol` policy entry is not used in `exec_attr`

Differences Between Trusted Solaris 8 Software and Solaris Trusted Extensions

The following sections summarize the components that remain, the components that have changed, and the components that have been removed in the change from Trusted Solaris to Solaris Trusted Extensions software.

Audit Events and Classes in Trusted Extensions

In Trusted Extensions, the audit classes for X events have been collapsed from six classes to four classes. The `xa` class and the `xl` class are removed. Events that were assigned to the `xa` class are in the `ot` class. Events that were assigned to the `xl` class are in the `lo` class. The bit masks of the remaining X audit classes have been changed from their Trusted Solaris 8 masks.

```
0x00800000:xc:X - object create/destroy
0x00400000:xp:X - privileged/administrative operations
0x01000000:xs:X - operations that always silently fail, if bad
0x01c00000:xx:X - all X events (meta-class
```

Device Management in Trusted Extensions

In Trusted Extensions, the `allocate` and `deallocate` commands are only available to TCB (Trusted Computing Base) processes that run in the global zone. Ordinary users must use the Device Manager GUI to allocate and deallocate devices.

Trusted Extensions device policy uses the Solaris `getdevpolicy` and `update_drv` interfaces. The Trusted Solaris 8 device policies: `data_mac_policy`, `attr_mac_policy`, `open_priv`, and `str_type` have been removed.

Files and File System Mounting in Trusted Extensions

Trusted Extensions provides no explicit mount attributes for specifying labels. The label of a mounted filesystem is the same as the label that is associated with the owning host or owning zone. Writing up is not permitted. Writing up is prevented by disallowing mounts of higher-labeled or disjointly labeled filesystems. Reading down is permitted. Reading down is enforced by restricting mounts of lower-labeled filesystems to be read-only.

The Trusted Extensions implementation for specifying security attributes on file systems follows the Solaris implementation. Therefore, files do not have forced privileges or allowed privileges. This implementation enables Trusted Extensions to support any file system that is supported by Solaris zones.

File relabeling is implemented by moving a file from one mounted file system to another file system.

Labels in Trusted Extensions

As in the Trusted Solaris releases, Trusted Extensions provides a `label_encodings` file. Labels, label ranges, clearances, and defaults are defined in the `label_encodings` file.

In Trusted Extensions, the `label_encodings` file that is installed by default defines commercial labels, such as `RESTRICTED` and `PUBLIC`. In Trusted Solaris releases, the default label encodings file, `label_encodings.multi`, was a version of a U.S. Government encodings file.

In the Label Builder, labels are shown in long form instead of in short form. When choosing a session clearance or workspace label, `Trusted Path` is used instead of `Admin Low` or `Admin High`.

Label APIs in Trusted Extensions

In Solaris Trusted Extensions, the label APIs that showed the internals of a label's structure are now obsolete. These label APIs have been replaced by the `label_to_str()` and `str_to_label()` functions. For the interfaces that are obsolete, and their replacement functions, see Table 7.

Also, CMW labels have been replaced by sensitivity labels. All CMW and IL (information label) interfaces have been removed.

Mail in Trusted Extensions

In the Solaris Trusted Extensions release, each zone has an independent instance of `sendmail`. Therefore, mail cannot be upgraded. Users can send mail and can receive mail only at the label of the user's workspace.

LDAP Naming Service in Trusted Extensions

Solaris Trusted Extensions uses LDAP as a naming service. In Trusted Extensions, NIS and NIS+ do not support the `tnrhdb` and `tnrhtp` databases. These naming services do not have a proxy server that can bind to a multilevel port (MLP). Therefore, the trusted networking databases cannot be reached from multiple zones concurrently.

Except for user passwords, LDAP data is considered public information. Therefore, any information in LDAP is not protected by a MAC policy. Instead, as in the Solaris OS, data is protected by an administrative policy. LDAP administrative policy is based on LDAP identities and passwords. When sensitivity labels are assigned as attributes of users and network endpoints, the labels are stored in an internal format. This format does not disclose classified information.

When an LDAP server is deployed as the naming service within a Trusted Extensions environment, the server must be configured to bind to a multilevel port (MLP) in the global zone.

Trusted Extensions can also be configured to rely on an existing LDAP infrastructure. In this case, an LDAP proxy server must be installed. This proxy server must be configured to bind to an MLP in the global zone of a system that is configured with Trusted Extensions. This Trusted Extensions system can then proxy multilevel requests from other zones and other hosts to the existing unlabeled LDAP server. The unlabeled server must be assigned the `admin_low` template in the `tnrhdb` of the proxy server.

To migrate NIS+ tables to LDAP entries, see the following man pages:

- `nisldapmaptest(1M)`
- `rpc.nisd(1M)`
- `rpc.nisd(4)`
- `NIS+LDAPmapping(4)`

Named Pipes in Trusted Extensions

In the Solaris OS, named pipes are used as one-way conduits. In Trusted Extensions, named pipes permit write-up operations. The writer runs at a lower label than the reader's dominant label. In Trusted Solaris 8, named pipes were configured by upgrading the label of the FIFO to the reader's label. In Trusted Extensions, named pipes are configured by using read-only lofs mounts of directories in lower-level zones into dominant higher-level zones. The FIFO is created at the label of the zone of the writer. For more information, see the `mkfifo(1M)` man page.

Networking in Trusted Extensions

Trusted Extensions does not support the TSIX or TSOL networking protocols. Trusted Extensions defines CIPSO-labeled templates and unlabeled templates in the `tnrhtp` database. The label `ADMIN_HIGH` is used as an upper bound, but is never transmitted as a CIPSO label. For more information, see "Zones in Trusted Extensions" on page 18.

The format of the `tnrhtp` database has been simplified because process attributes like privileges, user ids, and group ids are no longer supported. The format of the `tnrhdb` database is unchanged. The `tnzonecfg` database replaces the `tnidb` database, although the two databases are not equivalent.

The `/etc/security/tsol/tnrhtp` file that is installed with the Solaris Trusted Extensions release contains templates that can be used with any `label_encodings` file. The following table shows the correspondences between earlier versions of `tnrhtp` and the version that is shipped with the Solaris Trusted Extensions release.

TABLE 1 Template Names in the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris Template Name	Trusted Extensions Name	Note
`cipso`	`cipso`	For labeled hosts
`unlab`	`admin_low`	For unlabeled hosts
`tsol`, `tsol_cipso`, `tsix`	None	Use `cipso` template
`tsol_ripso`, `ripso_top_secret`	None	Removed

Network communication is restricted by label. By default, zones cannot communicate with each other because their labels are different.

Packets from unlabeled hosts that originate outside a Trusted Extensions domain can be labeled for trusted routing through the secure domain to another host outside the domain by using IP options. Incoming packets are labeled according to their originating host's entry in the `tnrhdb`. Incoming packets are routed through the Trusted Extensions domain according to their sensitivity level and the trusted routing information. The sensitivity label is still carried in the IP option. The label is stripped when the packet exits the trusted domain. IPv6 now supports trusted routing.

Dynamic routing is not supported. Static routing is supported.

Packaging in Trusted Extensions

Trusted Extensions software does not require special packaging attributes. Therefore, the `tsolinfo` file is no longer used.

PAM in Trusted Extensions

The PAM module for Trusted Extensions, `pam_tsol_account.so.1`, has only one module type and one function. The module is of type `account`, and the function checks the label range. The module has no options. No other Trusted Extensions-specific functions of PAM from Trusted Solaris 8 software are included in this release.

- If a PAM stack for `account` in the Trusted Solaris 8 release did not have `label_check_on` in `pam_tsol.so.1`, then you do not need to add `pam_tsol_account.so.1` to the corresponding stack in the Solaris Trusted Extensions release.
- If a PAM stack for `account` in the Trusted Solaris 8 release did have `label_check_on` in `pam_tsol.so.1`, then the corresponding stack in the Solaris Trusted Extensions release should use `pam_tsol_account.so.1` in the same place in the stack with no switches.

Trusted Extensions adds the `allow_unlabeled` option to PAM services. Together with the `allow_remote` option, administrators can manage headless systems remotely. For details, see the `pam_roles(5)` and `pam_tsol_account(5)` man pages.

PAM stacks for other module types should be used in the same manner for Trusted Extensions as for the Solaris OS. For more information, see the `pam(3PAM)` and `pam.conf(4)` man pages.

Policy in Trusted Extensions

In Trusted Extensions, a process' clearance is the same as its sensitivity label. Write up is not supported.

There is no administrative distinction between `ADMIN_HIGH` and `ADMIN_LOW` workspaces. Therefore, such workspaces are displayed as *Trusted Path*.

The `tsol` policy in the `exec_attr` file is removed. Use the `solaris` policy.

Printing in Trusted Extensions

Trusted Extensions supports both single-level and multilevel printing. Multilevel printing is implemented in the global zone only. The global zone must have its own IP address to be a multilevel print service. To use the global zone's print server, a labeled zone must have a separate IP address from the global zone.

Only multilevel printers have a label range. A printer's label range can be restricted with the Device Allocation Manager.

In Trusted Solaris releases, banner and trailer pages were enabled by default. In Trusted Extensions, administrators run a printer model script to add banner and trailer pages with security information to a printer.

`lpadmin -p` *printer* `-m` *printer-model-script*

Trusted Extensions adds four printer model scripts: `tsol_standard`, `tsol_netstandard`, `tsol_standard_foomatic`, and `tsol_netstandard_foomatic`.

Solaris Management Console in Trusted Extensions

The Solaris Management Console is no longer a multilevel service. The Solaris Management Console can only be contacted by clients that are running at the same label as the server. For most Trusted Extensions administration, access to the global zone is required. Because ordinary users are not permitted to log in to the global zone, only roles that are cleared for all labels can connect to the Solaris Management Console in the global zone.

Window System and CDE in Trusted Extensions

The login sequence is slightly different, and a new dialog box, Last Login, contains security information for the login user. The Shutdown menu item has been replaced with the Suspend System menu item, which checks for user authorization, then runs the `sys-suspend` command.

The `System_Admin` folder has been renamed to the `Trusted_Extensions` folder.

The CDE actions in the `Trusted_Extensions` folder have been updated. The NIS+ actions have been removed. Actions for administering LDAP and labeled zones have been added.

Zones in Trusted Extensions

Trusted Extensions uses zones for labeling. The global zone is an administrative zone, so is not available to users. The global zone is multilevel. The networking label of the global zone is `ADMIN_LOW`, but its process label is `ADMIN_HIGH`. Files that are private to the global zone are also labeled `ADMIN_HIGH`. Files that are shared with all zones are labeled `ADMIN_LOW`.

Each non-global zone has a unique label. Non-global zones are called *labeled zones*. Labeled zones are available to ordinary users. The global zone is available to roles only.

The Trusted Extensions policy for zones is different from Solaris policy. Trusted Extensions does not require a separate IP address per zone. However, all zones must have a single naming service. A single naming service provides all zones with a single set of users, UIDs, and GIDs.

Network communication is restricted by label. By default, zones cannot communicate with each other because their labels are different. The `/export` directory of a zone can be read by any zone whose label dominates the label of the `/export` directory.

Only system processes and roles are allowed to execute in the global zone. In certain cases, privileged processes in the global zone can be exempt from aspects of MAC policy. For example, system processes and roles that have the `file_dac_search` privilege and the `file_dac_read` privilege can access files which belong to labeled zones.

Privileges in Trusted Extensions

Privileges in Trusted Extensions are coded to correspond to their Solaris counterparts. Privileges in Solaris software are implemented differently from privileges in previous Trusted Solaris releases.

- Basic privileges are implemented. For example, `proc_exec` and `proc_info` are basic privileges.

  Basic privileges do not override security policy, but rather enable use of the system. Without the `proc_exec` privilege, a user cannot use the system.
- Privileges are not file attributes. Therefore, there are no allowed or forced privileges.
- Default and limit privileges can be assigned to the initial shell of a user or of a role.
- Privileges are called by name, not by number.

  Therefore, privilege numbers are not used in function calls or in the `exec_attr` file.
- Privilege macros are not used and have been removed.
- Privileges interact with zones. Some privileges can be used in the global zone only, so are not available to ordinary users.

For correspondences between Trusted Solaris privileges and Trusted Extensions privileges, see Table 1 in Interface Changes in the Solaris Trusted Extensions Release, Table 10, and "New Interfaces in Trusted Extensions Software" on page 24. For a complete list of privileges, see the `privileges(5)` man page.

The Solaris Trusted Extensions release adds the following privileges:

- `net_bindmlp` – Allows a process to bind to multilevel ports.
- `net_mac_aware` – Allows a process to communicate with peers at labels that are different from its own.

The Trusted Solaris command `runpd` has been replaced by the Solaris `ppriv -d` command. For details, see the `ppriv(1)` man page. For examples, see "How to Determine Which Privileges a Program Requires" in *System Administration Guide: Security Services*.

Trusted Extensions User Commands

On a system that is configured with Trusted Extensions, most Solaris user commands work as the commands work in the Solaris OS. Some command options apply to Trusted Extensions software only. Trusted Extensions also adds user commands. For a complete list, see "New Interfaces in Trusted Extensions Software" on page 24, Table 2, and Table 3.

Trusted Extensions System Administration Commands

On a system that is configured with Trusted Extensions, system administration commands work as follows:

- Most Solaris system administration commands work as the commands work in the Solaris OS, for example, `add_drv` and `share`.
- Some command options apply to Trusted Extensions software only, such as the `-R` option to `netstat`.
- Because NIS+ is not a supported naming service for a Trusted Extensions environment, NIS+ administration commands are not modified for this release.
- Some commands that are familiar to a Trusted Solaris 8 administrator have been modified, such as `chk_encodings`. For the changes, see the man pages.

  For links to the man pages, see Table 4 and "New Interfaces in Trusted Extensions Software" on page 24.

Trusted Extensions System Calls

On a system that is configured with Trusted Extensions, most Trusted Solaris system calls have been replaced by Solaris system calls. Some system calls are extended in Trusted Extensions software. For a complete list, see Table 5 and "New Interfaces in Trusted Extensions Software" on page 24.

Trusted Extensions Library Functions

On a system that is configured with Trusted Extensions, some functions have been modified. Some changes are due to architectural changes in the product. Some changes are due to removal of nonstandard interfaces.

The library functions for privileges that were provided by Trusted Solaris software have been replaced by Solaris functions. Label functions that manipulate CMW labels have been removed. Some label functions have been changed to make label structures opaque. Other label functions have been replaced by new label functions that make label structures opaque. Customers are encouraged to use the new interfaces when developing label-aware code for their sites.

For a complete list, see Table 6 and "New Interfaces in Trusted Extensions Software" on page 24.

Trusted Extensions Databases and Files

Databases and files have been reformatted to correspond to technical changes. Unneeded files have been removed. For the list, see Table 9 and "New Interfaces in Trusted Extensions Software" on page 24.

Trusted Extensions Devices and Drivers

On a system that is configured with Trusted Extensions, all Trusted Solaris device interfaces, and kernel functions for drivers have been replaced by Solaris functions. For the list, see Table 11.

Differences Between Solaris 10 8/07 Software and Solaris Trusted Extensions

Trusted Extensions builds on Solaris software, and can restrict the use of some Solaris utilities. The differences affect users, administrators, and developers. Configuration options that are optional on a Solaris system can be required by Trusted Extensions. For example, roles are required to administer the system, and the Solaris Management Console is required to administer users, roles, profiles, and the network. Zones must be installed, and each zone must be assigned a unique label.

Installation and Configuration of Trusted Extensions

Solaris Trusted Extensions installs as a set of packages on a newly installed Solaris 10 system. The following installation practices should be followed:

- Earlier Trusted Solaris releases cannot be upgraded to the current release.
- The software cannot be installed on a Solaris system that is already configured with non-global zones.

Desktops in Trusted Extensions

Solaris Trusted Extensions supports a trusted version of the Sun Java Desktop System, (Trusted JDS) as well as CDE. The Trusted CDE desktop continues to support the visible Trusted Solaris features, such as labels, trusted stripe, the Device Allocation Manager, the Admin Editor, and so on.

New administrative actions in CDE 1.7 are modified for security on the Trusted Extensions desktop. Actions that are unique to Trusted Extensions are in the `Trusted_Extensions` folder.

- The Style Manager should not be run from the Application Manager when Trusted Extensions is configured, because the Style Manager requires the trusted path. Run the Style Manager from the Front Panel and the Workspace menu, where the Style Manager has the trusted path.
- The contents of the `Trusted_Extensions` folder in the Application Manager has changed. Actions to administer zones have been added. NIS+ actions have been removed.
- As in the Trusted Solaris 8 2/04 release, the CDE Workspace Menu can be customized to add actions. For details, see "How to Customize the CDE Workspace Menu" in *Solaris Trusted Extensions User's Guide*.

Security Attributes on CDE Actions in Trusted Extensions Software

Trusted Extensions adds CDE actions to the objects that can be assigned security attributes in the `exec_attr` database. CDE actions can be constrained by label by customizing the Workspace Menu to include only actions that are relevant to a specific label. To customize the menu, see "How to Customize the CDE Workspace Menu" in *Solaris Trusted Extensions User's Guide*

Administration Tools in Trusted Extensions

Secure administration requires the use of GUIs that Trusted Extensions provides. Trusted Extensions provides actions in the `Trusted_Extensions` folder in CDE, a Device Allocation Manager, and the Solaris Management Console. Trusted Extensions adds tools and options to existing tools in the Solaris Management Console GUI. This GUI enables administrators to manage users, networks, zones, and other databases. After launching the Solaris Management Console, the administrator chooses a Trusted Extensions "toolbox". The toolbox is a collection of programs. The administrator then uses the programs that are permitted to the role.

Trusted Device Management

The Solaris OS provides three methods of managing devices: the Volume Manager (`vold`), `logindevperm` and device allocation. As in the Trusted Solaris 8 releases, Trusted Extensions supports only device allocation. The Device Allocation Manager GUI is used to create an allocatable device. All devices that are allocated to a zone get deallocated when that zone shuts down, halts, or reboots. Device allocation can be done remotely or in shell scripts only from the global zone.

The `allocate`, `deallocate`, and `list_devices` commands do not work in labeled zones for roles or ordinary users. Users and roles must use the Device Allocation Manager GUI to allocate, deallocate and list devices. Trusted Extensions adds the `solaris.device.config` authorization to configure devices.

Trusted Printing

To manage printers, use the Printer Administrator action in the `System_Admin` folder in the global zone. To limit the label range of a printer, use the Device Allocation Manager in the global zone.

Trusted Extensions Software and Removable Media

Use the Solaris Management Console Devices and Hardware tool to manage serial lines and serial ports in the global zone. To limit the label range of removable media, use the Device Allocation Manager in the global zone.

Additional Rights and Authorizations in Trusted Extensions

The Solaris Trusted Extensions release adds privileged commands to the Device Security profile, and privileged actions to many profiles.

The Solaris Trusted Extensions release adds the following authorizations:

- `solaris.file.`
- `solaris.label.`
- `solaris.print.`
- `solaris.smf.manage.labels`
- `solaris.smf.manage.tnctl`
- `solaris.smf.manage.tnd`
- `solaris.smf.value.tnd`

The Solaris Trusted Extensions release adds the following rights profiles:

- All Actions
- Basic Actions
- Information Security
- Object Label Management
- Outside Accred

The Solaris Trusted Extensions release adds label authorizations and service management authorizations to the following rights profiles:

- Maintenance and Repair
- Printer Management
- User Security
- Network Management
- Network Security

Together, the Information Security and the User Security rights profiles define the Security Administrator role.

New Interfaces in Trusted Extensions Software

The new interfaces in the Solaris Trusted Extensions release are listed in the following table by man page section number. The table includes some Solaris interfaces that perform critical functions for Trusted Extensions.

Only interfaces whose names have changed are included in the table. However, interfaces whose names have not changed might have different options or different functionality in this release. For a complete list, see Interface Changes in the Solaris Trusted Extensions Release.

TABLE 2 New Man Pages in Solaris Trusted Extensions Software

Man Page	Note
`getzonepath`(1)	Replaces `getsldname`.
`ldaplist`(1)	Trusted Extensions network databases are added to the LDAP directory server.
`ppriv`(1)	Solaris command replaces Trusted Solaris commands that handled privileges.
`smtnzonecfg`(1M)	Manages trusted network zone configuration database.
`getpflags`(2)	Trusted Extensions adds the `NET_MAC_AWARE` flag.
`getlabel`(2)	Gets sensitivity label of file.
`setpflags`(2)	Trusted Extensions adds the `NET_MAC_AWARE` flag.
`is_system_labeled`(3C)	Determines if the system is configured with Trusted Extensions.
`getpeerucred`(3C)	Works as in Solaris OS. Replaces `getpeerinfo()`.
`priv_gettext`(3C)	Works as in Solaris OS. Replaces `get_priv_text()`.
`ucred_getlabel`(3C)	`ucred_getlabel()` reads the label on a process.
`libtsnet`(3LIB)	Describes the `libtsnet()` interfaces.

TABLE 2 New Man Pages in Solaris Trusted Extensions Software *(Continued)*

Man Page	Note
libtsol(3LIB)	Describes the `libtsol()` interfaces.
getdevicerange(3TSOL)	Gets the label range of a device.
getpathbylabel(3TSOL)	Gets the full pathname. Replaces `mldrealpathl()`.
getplabel(3TSOL)	Gets the sensitivity label of a process.
getuserrange(3TSOL)	Gets the label range of a user.
getzoneidbylabel(3TSOL)	Gets the ID of a zone.
getzonelabelbyid(3TSOL)	Gets the label of a zone.
getzonelabelbyname(3TSOL)	
getzonerootbyid(3TSOL)	Gets the full pathname of a zone.
getzonerootbylabel(3TSOL)	
getzonerootbyname(3TSOL)	
label_to_str(3TSOL)	Converts labels to strings. Replaces `bcltobanner()` and other interfaces.
m_label(3TSOL)	`m_label()` is a placeholder for the allocation, duplication, and free functions.
m_label_alloc(3TSOL)	Manages storage for opaque labels.
m_label_dup(3TSOL)	Duplicates a label.
m_label_free(3TSOL)	Frees storage for opaque labels.
setflabel(3TSOL)	Replaces `setcmwlabel()`.
str_to_label(3TSOL)	Converts labels to strings. Replaces `stobsl()` and `stobclear()`.
tsol_getrhtype(3TSOL)	Gets the host type of the specified hostname.
door_ucred(3DOOR)	Works as in Solaris OS. Replaces `door_tcred()`.
getsockopt(3SOCKET)	Trusted Extensions adds the `SO_MAC_EXEMPT` option.
getsockopt(3XNET)	
setsockopt(3SOCKET)	
setsockopt(3XNET)	
tnzonecfg(4)	Is the local configuration file for the global zone and labeled zones.
TrustedExtensionsPolicy(4)	Is the policy file for window behavior. Replaces `config.privs`.
labels(5)	Describes label policy.

TABLE 2 New Man Pages in Solaris Trusted Extensions Software *(Continued)*

Man Page	Note
pam_tsol_account(5)	Is the PAM module for account authentication.
privileges(5)	Contains descriptions of new privileges, net_bindmlp and net_mac_aware.

◆ ◆ ◆

Interface Changes in the Solaris Trusted Extensions Release

Many interfaces changed between the Trusted Solaris 8 releases and the current release. For a list of new interfaces, see "New Interfaces in Trusted Extensions Software" on page 24.

The following tables list every interface change, the current man page, and the reason for the change. Changes can be one of the following:

- **No change –** The interface is originally a Trusted Solaris interface, and has not significantly changed for this release.
- **Remains unsupported –** Due to security issues, the interface is not supported in Solaris Trusted Extensions software.
- **Removed** - Due to changes in the architecture, the interface was removed. A `None` entry in the second column indicates a removed interface. The Note column contains `Removed` or an explanation.
- **Replaced –** The interface is renamed.
- **Has new options** - For a Solaris interface, the interface provides options that are specific to Trusted Extensions software. For an interface that is unique to Trusted Extensions, the Solaris Trusted Extensions options are different from the options in Trusted Solaris software.
- **Obsolete –** A Trusted Solaris interface has been replaced with a Trusted Extensions interface with a new name and new parameters. Many label interfaces have been replaced with interfaces that hide the internals of the label. Although recompiled programs that use the obsolete interfaces might work, programs and scripts should be written to use the new interfaces.
- **Support removed –** Support for a feature, such as CMW labels, was removed entirely, or removed from this interface.
- **Use** *interface* **–** Use an equivalent Solaris interface.
- **Works as in Solaris OS –** Due to changes in the architecture, Solaris Trusted Extensions software does not modify the Solaris OS interface.

TABLE 1 Privilege Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Privilege Identifier in Trusted Solaris 8		Current Privilege Name	Note
1	FILE_AUDIT	None	Removed.
2	FILE_CHOWN	FILE_CHOWN, FILE_CHOWN_SELF	
3	FILE_DAC_EXECUTE	FILE_DAC_EXECUTE	
4	FILE_DAC_READ	FILE_DAC_READ	
5	FILE_DAC_SEARCH	FILE_DAC_SEARCH	
6	FILE_DAC_WRITE	FILE_DAC_WRITE	
8	FILE_DOWNGRADE_SL	FILE_DOWNGRADE_SL	
9	FILE_LOCK	None	Removed.
10	FILE_MAC_READ	None	Removed.
11	FILE_MAC_SEARCH	None	Removed.
12	FILE_MAC_WRITE	None	Removed.
14	FILE_OWNER	FILE_OWNER	
15	FILE_SETDAC	None	Removed.
16	FILE_SETID	FILE_SETID	
17	FILE_SETPRIV	None	Removed.
19	FILE_UPGRADE_SL	FILE_UPGRADE_SL	
20	IPC_DAC_READ	IPC_DAC_READ	
21	IPC_DAC_WRITE	IPC_DAC_WRITE	
23	IPC_MAC_READ	None	Removed.
24	IPC_MAC_WRITE	None	Removed.
26	IPC_OWNER	IPC_OWNER	
30	NET_BROADCAST	None	Removed.
32	NET_DOWNGRADE_SL	None	Removed.
33	NET_MAC_READ	NET_MAC_AWARE, NET_BINDMLP	Slightly different privilege that enforces current security architecture.
35	NET_PRIVADDR	NET_PRIVADDR	
36	NET_RAWACCESS	NET_RAWACCESS	

TABLE 1 Privilege Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Privilege Identifier in Trusted Solaris 8		Current Privilege Name	Note
37	NET_REPLY_EQUAL	NET_MAC_AWARE	Slightly different privilege that enforces current security architecture.
38	NET_SETCLR	None	Removed.
39	NET_SETID	None	Removed.
40	NET_SETPRIV	None	Removed.
42	NET_UPGRADE_SL	None	Removed.
43	PROC_AUDIT_APPL	PROC_AUDIT	
44	PROC_AUDIT_TCB	PROC_AUDIT	
45	PROC_CHROOT	PROC_CHROOT	
46	PROC_DUMPCORE	None	Removed.
47	PROC_MAC_READ	PROC_ZONE	Provides equivalent functionality.
48	PROC_MAC_WRITE	PROC_ZONE	Provides equivalent functionality.
49	PROC_NODELAY	None	Removed.
51	PROC_OWNER	PROC_OWNER	
52	PROC_SETCLR	None	Removed.
53	PROC_SETID	PROC_SETID	
55	PROC_SETSL	None	Removed.
29	PROC_DEBUG_NONTRANQUIL	None	Removed.
56	PROC_TRANQUIL	None	Removed.
57	SYS_AUDIT	SYS_AUDIT	
58	SYS_BOOT	None	Removed.
59	SYS_CONFIG	SYS_CONFIG	
60	SYS_CONSOLE	None	Removed.
61	SYS_DEVICES	SYS_DEVICES	
63	SYS_FS_CONFIG	None	Removed.
64	SYS_IPC_CONFIG	SYS_IPC_CONFIG	
65	SYS_MAXPROC	SYS_RESOURCE	

TABLE 1 Privilege Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Privilege Identifier in Trusted Solaris 8		Current Privilege Name	Note
66	SYS_MINFREE	SYS_RESOURCE	
67	SYS_MOUNT	SYS_MOUNT	
68	SYS_NET_CONFIG	SYS_NET_CONFIG	
69	SYS_NFS	SYS_NFS	
70	SYS_SUSER_COMPAT	SYS_SUSER_COMPAT	
28	SYS_SYSTEM_DOOR	None	Removed.
71	SYS_TRANS_LABEL	SYS_TRANS_LABEL	
72	WIN_COLORMAP	WIN_COLORMAP	
73	WIN_CONFIG	WIN_CONFIG	
74	WIN_DAC_READ	WIN_DAC_READ	
75	WIN_DAC_WRITE	WIN_DAC_WRITE	
76	WIN_DGA	WIN_DGA	
77	WIN_DEVICES	WIN_DEVICES	
79	WIN_DOWNGRADE_SL	WIN_DOWNGRADE_SL	
80	WIN_FONTPATH	WIN_FONTPATH	
81	WIN_MAC_READ	WIN_MAC_READ	
82	WIN_MAC_WRITE	WIN_MAC_WRITE	
84	WIN_SELECTION	WIN_SELECTION	
86	WIN_UPGRADE_SL	WIN_UPGRADE_SL	

TABLE 2 User Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris 8 Command	Current Man Page	Note
adornfc	None	Files are no longer adorned.
allocate	allocate(1)	Has new options.
at	at(1)	Works as in Solaris OS.
atq	atq(1)	Works as in Solaris OS.
atrm	atrm(1)	Works as in Solaris OS.

TABLE 2 User Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
batch	at(1)	Works as in Solaris OS.
cancel	cancel(1)	Caller must be equal to cancel job.
chgrp	chgrp(1)	Works as in Solaris OS.
chmod	chmod(1)	Works as in Solaris OS.
chown	chown(1)	Works as in Solaris OS.
crle	crle(1)	Works as in Solaris OS.
crontab	crontab(1)	Works as in Solaris OS.
date	date(1)	Works as in Solaris OS.
deallocate	deallocate(1)	Has new options.
disable	disable(1)	Works as in Solaris OS.
dtappsession	dtappsession(1)	No change.
du	du(1)	Works as in Solaris OS.
enable	enable(1)	Works as in Solaris OS.
find	find(1)	Works as in Solaris OS.
getfattrflag	None	Files no longer have flags.
getfpriv	None	Files no longer have forced privileges.
getlabel	getlabel(1)	Gets the sensitivity label of the containing file system.
getmldadorn	None	Files are no longer adorned.
getsldname	getzonepath(1)	Renamed. Gets the zone root of the file.
ipcrm	ipcrm(1)	Works as in Solaris OS.
ipcs	ipcs(1)	Works as in Solaris OS.
kbd	kbd(1)	Works as in Solaris OS.
ld	ld(1)	Works as in Solaris OS.
list_devices	list_devices(1)	Has new options.
login	login(1)	Works as in Solaris OS.
lp	lp(1)	Extended to require authorization for suppressing banners and labels.

TABLE 2 User Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
`lpstat`	`lpstat(1)`	Except for listing, caller must dominate the job's label. For listing, caller must equal the job's label.
`mkdir`	`mkdir(1)`	Works as in Solaris OS.
`mldpwd`	None	MLDs and SLDs are removed.
`mldrealpath`	None	MLDs and SLDs are removed.
`nca`	`nca(1)`	Remains unsupported.
`ncakmod`	`ncakmod(1)`	Remains unsupported.
`nispasswd`	None	NIS+ is not a supported name service for Solaris Trusted Extensions software. For LDAP information, see `ldap(1)`.
`passwd`	`passwd(1)`	Works as in Solaris OS.
`pattr`	None	Processes no longer have attribute flags.
`pclear`	None	No longer needed.
`pcred`, `pfiles`, `pflags`	`pcred(1)`	Work as in Solaris OS.
`pfsh`	`pfsh(1)`	Works as in Solaris OS.
`plabel`	`plabel(1)`	Only returns the sensitivity label.
`pldd`, `pmap`	`pldd(1)`	Work as in Solaris OS.
`ppriv`	`ppriv(1)`	Works as in Solaris OS.
`pprivtest`	`ppriv(1)`	Use the `ppriv -D` command.
`proc`	`proc(1)`	Works as in Solaris OS.
`profiles`	`profiles(1)`	Works as in Solaris OS.
`prun`, `psig`, `pstack`, `pstop`, `ptime`, `ptree`, `pwait`, `pwdx`	`proc(1)`	Work as in Solaris OS.
`rm`	`rm(1)`	Works as in Solaris OS.
`rmdir`	`rmdir(1)`	Works as in Solaris OS.
`roles`	`roles(1)`	Works as in Solaris OS.
`setfattrflag`	None	Files no longer have flags.
`setfpriv`	None	Files no longer have forced privileges.

TABLE 2 User Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
`setlabel`	`setlabel(1)`	Files are moved between labeled zones.
`snca`	`nca(1)`	Remains unsupported.
`tar`	`tar(1)`	Has new options.
`testfpriv`	None	Files no longer have forced privileges.
`tfind`	None	MLDs and SLDs are removed.
`uname`	`uname(1)`	Works as in Solaris OS.
`vacation`	`vacation(1)`	Works as in Solaris OS.

TABLE 3 User Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases – 1b Interfaces

Trusted Solaris 8 Command	Current Man Page	Note
`lpc`	`lpc(1B)`	Works as in Solaris OS.
`lpq`	`lpq(1B)`	Caller must dominate label of jobs.
`lpr`	`lpr(1B)`	Works as in Solaris OS.
`lprm`	`lprm(1B)`	Caller must dominate label of job.

TABLE 4 System Administration Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris 8 Command	Current Man Page	Note
`add_allocatable`	`add_allocatable(1M)`	Has new options.
`add_drv`	`add_drv(1M)`	Works as in Solaris OS.
`adminvi`		Removed.
`arp`	`arp(1M)`	Works as in Solaris OS.
`atohexlabel`	`atohexlabel(1M)`	CMW support removed. Returns label in shortened hexadecimal format.
`audit`	`audit(1M)`	Works as in Solaris OS.
`audit_startup`	`audit_startup(1M)`	Works as in Solaris OS.
`audit_warn`	`audit_warn(1M)`	Works as in Solaris OS.
`auditconfig`	`auditconfig(1M)`	Adds Trusted Extensions policies.

TABLE 4 System Administration Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
auditd	auditd(1M)	Works as in Solaris OS.
auditreduce	auditreduce(1M)	Adds -l option.
auditstat	auditstat(1M)	Works as in Solaris OS.
automount	automount(1M)	Extended to automount home directories in labeled zones.
automountd	automountd(1M)	Works as in Solaris OS.
autopush	autopush(1M)	Works as in Solaris OS.
bootparamd	bootparamd(1M)	Works as in Solaris OS.
bsmconv, bsmunconv	bsmconv(1M)	Works as in Solaris OS.
chk_encodings	chk_encodings(1M)	Enhanced to check for CIPSO labels. Adds the -a option to provide a detailed report.
chroot	chroot(1M)	Works as in Solaris OS.
coreadm	coreadm(1M)	Works as in Solaris OS.
cron	cron(1M)	Works as in Solaris OS.
devfsadm	devfsadm(1M)	Works as in Solaris OS.
devfsadmd	devfsadmd(1M)	Works as in Solaris OS.
device_clean	device_clean(5)	Has new options. Change in man page section.
devpolicy	getdevpolicy(1M)	Works as in Solaris OS.
dfmounts	dfmounts(1M)	Works as in Solaris OS.
dfshares	dfshares(1M)	Works as in Solaris OS.
dispadmin	dispadmin(1M)	Works as in Solaris OS.
dminfo	dminfo(1M)	Works as in Solaris OS.
dl_booting	None	Removed in Solaris OS.
dl_restore	None	Removed in Solaris OS.
drvconfig	drvconfig(1M)	Works as in Solaris OS.
eeprom	eeprom(1M)	Works as in Solaris OS.
format	format(1M)	Works as in Solaris OS.

TABLE 4 System Administration Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
`fsdb_ufs`	`fsdb_ufs`(1M)	Works as in Solaris OS.
`ftpd`	`ftpd`(1M)	Works as in Solaris OS.
`fuser`	`fuser`(1M)	Works as in Solaris OS.
`getfsattr`, `getfsattr_ufs`	None	File system attributes have been removed.
`halt`	`halt`(1M)	Works as in Solaris OS.
`hextoalabel`	`hextoalabel`(1M)	CMW support removed.
`ifconfig`	`ifconfig`(1M)	Adds all-zones flag.
`in.ftpd`	`in.ftpd`(1M)	Works as in Solaris OS.
`in.named`	None	Removed in Solaris OS.
`in.rarpd`	`in.rarpd`(1M)	Works as in Solaris OS.
`in.rdisc`	`in.rdisc`(1M)	Works as in Solaris OS.
`in.rexecd`	`in.rexecd`(1M)	Works as in Solaris OS.
`in.rlogind`	`in.rlogind`(1M)	Works as in Solaris OS.
`in.routed`	`in.routed`(1M)	Works as in Solaris OS.
`in.rshd`	`in.rshd`(1M)	Works as in Solaris OS.
`in.tftpd`	`in.tftpd`(1M)	Works as in Solaris OS.
`inetd`	`inetd`(1M)	Works as in Solaris OS.
`init`	`init`(1M)	Works as in Solaris OS.
`init.wbem`	`init.wbem`(1M)	Works as in Solaris OS.
`install`	`install`(1M)	Works as in Solaris OS.
`ipsecconf`	`ipsecconf`(1M)	Works as in Solaris OS.
`ipseckey`	`ipseckey`(1M)	Works as in Solaris OS.
`lockd`	`lockd`(1M)	Works as in Solaris OS.
`lpadmin`	`lpadmin`(1M)	Caller must dominate to view, be equal to change a job.
`lpfilter`	`lpfilter`(1M)	Works as in Solaris OS.
`lpforms`	`lpforms`(1M)	Works as in Solaris OS.

TABLE 4 System Administration Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
`lpmove`	`lpmove`(1M)	Caller must be equal to move a job.
`lpsched`	`lpsched`(1M)	Caller must dominate to view, be equal to change the print service.
`lpshut`	`lpshut`(1M)	Works as in Solaris OS.
`lpsystem`	`lpsystem`(1M)	Works as in Solaris OS.
`lpusers`	`lpusers`(1M)	Works as in Solaris OS.
`mkdevalloc`	`mkdevalloc`(1M)	Works as in Solaris OS.
`mkdevdb`	None	Removed.
`mkdevmaps`	`mkdevmaps`(1M)	Works as in Solaris OS.
`modload`, `modunload`	`modload`(1M)	Works as in Solaris OS.
`mount`	`mount`(1M)	Works as in Solaris OS.
`mount_hsfs`	`mount_hsfs`(1M)	Works as in Solaris OS.
`mount_nfs`	`mount_nfs`(1M)	Works as in Solaris OS.
`mount_pcfs`	`mount_pcfs`(1M)	Works as in Solaris OS.
`mount_tmpfs`	`mount_tmpfs`(1M)	Works as in Solaris OS.
`mount_ufs`	`mount_ufs`(1M)	Works as in Solaris OS.
`mountall`	`mountall`(1M)	Works as in Solaris OS.
`mountd`	`mountd`(1M)	Works as in Solaris OS.
`named`	`named`(1M)	Works as in Solaris OS.
`netstat`	`netstat`(1M)	Adds `-R` option.
`newsecfs`	None	File system attributes have been removed.
`nfsd`	`nfsd`(1M)	Works as in Solaris OS.
`nfsstat`	`nfsstat`(1M)	Works as in Solaris OS.
All NIS+ commands: `nis_cachemgr`, `nisclient`, `nisd`, `nisd_resolv`, `nispasswdd`, `nispopulate`, `nissetup`, `nslookup`	None	NIS+ is no longer a supported naming service for Solaris Trusted Extensions software.
`nscd`	`nscd`(1M)	Works as in Solaris OS.

TABLE 4 System Administration Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
pbind	pbind(1M)	Works as in Solaris OS.
pkgchk	pkgchk(1M)	Works as in Solaris OS.
poweroff	poweroff(1M)	Works as in Solaris OS.
praudit	praudit(1M)	Works as in Solaris OS.
prtconf	prtconf(1M)	Works as in Solaris OS.
psradm	psradm(1M)	Works as in Solaris OS.
rarp	rarp(7P)	Works as in Solaris OS.
rdate	rdate(1M)	Works as in Solaris OS.
rdisc	rdisc(1M)	Works as in Solaris OS.
reboot	reboot(1M)	Works as in Solaris OS.
reject	reject(1M)	Works as in Solaris OS.
rem_drv	rem_drv(1M)	Works as in Solaris OS.
remove_allocatable	remove_allocatable(1M)	Has new options.
rexecd	rexecd(1M)	Works as in Solaris OS.
rlogind	rlogind(1M)	Works as in Solaris OS.
rmmount	rmmount(1M)	Works as in Solaris OS.
route	route(1M)	Adds -secattr option.
routed	routed(1M)	Works as in Solaris OS.
rpc.bootparamd	rpc.bootparamd(1M)	Works as in Solaris OS.
rpc.getpeerinfod	getpeerucred(3C)	Replaced with Solaris OS function.
rpc.nisd	None	NIS+ is no longer a supported naming service for Solaris Trusted Extensions software.
rpc.nisd_resolv	None	No longer supported.
rpc.nispasswdd	None	No longer supported.
rpc.tbootparamd	None	No longer needed.
rpc.yppasswdd	rpc.yppasswdd(1M)	Works as in Solaris OS.
rpc.ypupdated	rpc.ypupdated(1M)	Works as in Solaris OS.

TABLE 4 System Administration Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
rpcbind	rpcbind(1M)	Works as in Solaris OS.
rpcinfo	rpcinfo(1M)	Works as in Solaris OS.
rshd	rshd(1M)	Works as in Solaris OS.
runpd	ppriv(1)	Use `ppriv -d` command.
rwall	rwall(1M)	Works as in Solaris OS.
sendmail	sendmail(1M)	Works as in Solaris OS.
setaudit	auditconfig(1M)	Use the `-setaudit` option.
setfsattr	None	File system attributes have been removed.
setuname	setuname(1M)	Works as in Solaris OS.
share	share(1M)	Works as in Solaris OS.
share_nfs	share_nfs(1M)	Works as in Solaris OS.
shareall	shareall(1M)	Works as in Solaris OS.
showmount	showmount(1M)	Works as in Solaris OS.
smc	smc(1M)	Works as in Solaris OS.
smcron	smcron(1M)	Works as in Solaris OS.
smexec	smexec(1M)	Extended to support Trusted Extensions toolboxes and CDE actions.
smgroup	smgroup(1M)	Works as in Solaris OS.
smhost		Removed.
smmaillist	smmaillist(1M)	Works as in Solaris OS.
smmultiuser	smmultiuser(1M)	Works as in Solaris OS.
smnetidb	smtnzonecfg(1M)	Changed to work with current security architecture.
smnettmpl	smtnrhtp(1M)	Changed to work with current security architecture.
smnetwork	smtnrhdb(1M)	Changed to work with current security architecture.
smprofile	smprofile(1M)	Works as in Solaris OS.

TABLE 4 System Administration Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
`smrole`	`smrole`(1M)	Extended to support label information and Trusted Extensions toolboxes.
`smuser`	`smuser`(1M)	Extended to support label information and Trusted Extensions toolboxes.
`snoop`	`snoop`(1M)	Works as in Solaris OS.
`spray`	`spray`(1M)	Works as in Solaris OS.
`statd`	`statd`(1M)	Works as in Solaris OS.
`su`	`su`(1M)	Works as in Solaris OS.
`swap`	`swap`(1M)	Works as in Solaris OS.
`sysdef`	`sysdef`(1M)	Works as in Solaris OS.
`sysh`	`smf_method`(5)	Use the Solaris 10 service management framework.
`tbootparam`	None	No longer needed.
`telinit`	`telinit`(1M)	Works as in Solaris OS.
`tftpd`	`tftpd`(1M)	Works as in Solaris OS.
`tnchkdb`	`tnchkdb`(1M)	Has new options.
`tnctl`	`tnctl`(1M)	Has new options. Is now managed by the service management framework.
`tnd`	`tnd`(1M)	Has new options. Is now managed by the service management framework.
`tninfo`	`tninfo`(1M)	Has new options.
`tokmapctl`	None	TSIX functionality has been removed.
`tokmapd`	None	TSIX functionality has been removed.
`uadmin`	`uadmin`(1M)	Works as in Solaris OS.
`umount`	`umount`(1M)	Works as in Solaris OS.
`umountall`	`umountall`(1M)	Works as in Solaris OS.

TABLE 4 System Administration Command Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Command	Current Man Page	Note
`unshare`	`unshare`(1M)	Works as in Solaris OS.
`unshare_nfs`	`unshare_nfs`(1M)	Works as in Solaris OS.
`unshareall`	`unshareall`(1M)	Works as in Solaris OS.
`updatehome`	`updatehome`(1M)	No change.
`writeaudit`	None	Use Solaris OS audit interfaces.
`ypbind`	`ypbind`(1M)	Works as in Solaris OS.
`yppasswdd`	`yppasswdd`(1M)	Works as in Solaris OS.
`ypserv`	`ypserv`(1M)	Works as in Solaris OS.
`ypupdated`	`ypupdated`(1M)	Works as in Solaris OS.
`ypxfr`	`ypxfr`(1M)	Works as in Solaris OS.
`ypxfr_1perday`	`ypxfr_1perday`(1M)	Works as in Solaris OS.
`ypxfr_1perhour`	`ypxfr_1perhour`(1M)	Works as in Solaris OS.
`ypxfrd`	`ypxfrd`(1M)	Works as in Solaris OS.

TABLE 5 System Call Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris 8 System Call	Current Man Page	Note
`access()`	`access`(2)	Works as in Solaris OS.
`acct()`	`acct`(2)	Remains unsupported.
`acl()`	`acl`(2)	Works as in Solaris OS.
`adjtime()`	`adjtime`(2)	Works as in Solaris OS.
`audit()`	`audit`(2)	Works as in Solaris OS.
`auditon()`	`auditon`(2)	Adds Trusted Extensions policies.
`auditsvc()`	None	No longer a public interface.
`chdir()`	`chdir`(2)	Works as in Solaris OS.
`chmod()`	`chmod`(2)	Works as in Solaris OS.
`chown()`	`chown`(2)	Works as in Solaris OS.

TABLE 5 System Call Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 System Call	Current Man Page	Note
`chroot()`	`chroot(2)`	Works as in Solaris OS.
`chstate()`	None	Removed from Solaris OS.
`creat()`	`creat(2)`	Works as in Solaris OS.
`devpolicy()`	None	No longer needed.
`exec()`, `execl()`, `execle()`, `execlp()`, `execv()`, `execve()`, `execvp()`	`exec(2)`	Works as in Solaris OS.
`facl()`	`facl(2)`	Works as in Solaris OS.
`fchdir()`	`fchdir(2)`	Works as in Solaris OS.
`fchmod()`	`fchmod(2)`	Works as in Solaris OS.
`fchown()`	`fchown(2)`	Works as in Solaris OS.
`fchroot()`	`fchroot(2)`	Works as in Solaris OS.
`fgetcmwfsrange()`	None	File systems no longer have attributes.
`fgetcmwlabel()`	`fgetlabel(2)`	Replaced.
`fgetfattrflag()`	None	Files no longer have flags.
`fgetfpriv()`	None	Files no longer have forced privileges.
`fgetfsattr()`	None	File systems no longer have attributes.
`fgetmldadorn()`	None	Files are no longer adorned.
`fgetsldname()`	None	MLDs and SLDs are removed.
`fork()`, `fork1()`	`fork(2)`	Works as in Solaris OS.
`fpathconf()`	`fpathconf(2)`	Works as in Solaris OS.
`fsetcmwlabel()`	None	Files no longer have attributes.
`fsetfattrflag()`	None	Files no longer have flags.
`fsetfpriv()`	None	Files no longer have forced privileges.
`fstat()`	`fstat(2)`	Works as in Solaris OS.
`fstatvfs()`	`fstatvfs(2)`	Works as in Solaris OS.

TABLE 5 System Call Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 System Call	Current Man Page	Note
`getaudit()`, `getaudit_addr()`	`getaudit(2)`	Works as in Solaris OS.
`getauid()`	`getauid(2)`	Works as in Solaris OS.
`getclearance()`	None	In Solaris Trusted Extensions software, the process clearance is the same as the label.
`getcmwfsrange()`	None	File systems no longer have attributes.
`getcmwlabel()`	`getlabel(2)`	Replaced.
`getcmwplabel()`	`getplabel(3TSOL)`	Process label is zone's label.
`getdents()`	`getdents(2)`	Works as in Solaris OS.
`getfattrflag()`	None	Files no longer have flags.
`getfpriv()`	None	Files no longer have forced privileges.
`getfsattr()`	None	File systems no longer have attributes.
`getgroups()`	`getgroups(2)`	Works as in Solaris OS.
`getmldadorn()`	None	Files are no longer adorned.
`getmsgqcmwlabel()`	None	No longer needed.
`getpattr()`	`getpflags(2)`	Returns a different set of flags.
`getpgid()`, `getpgrp()`, `getpid()`, `getppid()`	`getpid(2)`	Works as in Solaris OS.
`getppriv()`	`getppriv(2)`	Works as in Solaris OS.
`getrlimit()`	`getrlimit(2)`	Works as in Solaris OS.
`getsemcmwlabel()`	None	No longer needed.
`getshmcmwlabel()`	None	No longer needed.
`getsid()`	`getsid(2)`	Works as in Solaris OS.
`getsldname()`	`getzonerootbyid(3TSOL)`, `getzonerootbylabel(3TSOL)`, `getzonerootbyname(3TSOL)`	MLDs and SLDs are replaced by zone root paths.
`kill()`	`kill(2)`	Works as in Solaris OS.

TABLE 5 System Call Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 System Call	Current Man Page	Note
`lchown()`	`lchown(2)`	Works as in Solaris OS.
`lgetcmwlabel()`	None	Files no longer have attributes.
`link()`	`link(2)`	Works as in Solaris OS.
`llseek()`	`llseek(2)`	Works as in Solaris OS.
`lseek()`	`lseek(2)`	Works as in Solaris OS.
`lsetcmwlabel()`	None	Files no longer have attributes.
`lstat()`	`lstat(2)`	Works as in Solaris OS.
`mkdir()`	`mkdir(2)`	Works as in Solaris OS.
`mknod()`	`mknod(2)`	Works as in Solaris OS.
`mldgetfattrflag()`	None	Files no longer have flags.
`mldsetfattrflag()`	None	Files no longer have flags.
`mount()`	`mount(2)`	Works as in Solaris OS.
`msgctl()`	`msgctl(2)`	Works as in Solaris OS.
`msgget()`	`msgget(2)`	Works as in Solaris OS.
`msggetl()`	None	No longer needed.
`msgrcv()`	`msgrcv(2)`	Works as in Solaris OS.
`msgsnd()`	`msgsnd(2)`	Works as in Solaris OS.
`nice()`	`nice(2)`	Works as in Solaris OS.
`open()`	`open(2)`	Works as in Solaris OS.
`p_online()`	`p_online(2)`	Works as in Solaris OS.
`pathconf()`	`pathconf(2)`	Works as in Solaris OS.
`pread()`	`pread(2)`	Works as in Solaris OS.
`preadl()`	None	No longer needed.
`priocntl()`	`priocntl(2)`	Works as in Solaris OS.
`priocntlset()`	`priocntlset(2)`	Works as in Solaris OS.
`processor_bind()`	`processor_bind(2)`	Works as in Solaris OS.
`pwrite()`	`pwrite(2)`	Works as in Solaris OS.

TABLE 5 System Call Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 System Call	Current Man Page	Note
`pwritel()`	None	No longer needed.
`read()`	`read(2)`	Works as in Solaris OS.
`readl()`	None	Removed.
`readlink()`	`readlink(2)`	Works as in Solaris OS.
`readv()`	`read(2)`	Works as in Solaris OS.
`readvl()`	None	No longer needed.
`rename()`	`rename(2)`	Works as in Solaris OS.
`rmdir()`	`rmdir(2)`	Works as in Solaris OS.
`secconf()`	None	Removed.
`semctl()`	`semctl(2)`	Works as in Solaris OS.
`semget()`	`semget(2)`	Works as in Solaris OS.
`semgetl()`	None	No longer needed.
`semop()`, `semtimedop()`	`semop(2)`	Works as in Solaris OS.
`setaudit()`, `setaudit_addr()`	`setaudit(2)`	Works as in Solaris OS.
`setauid()`	`setauid(2)`	Works as in Solaris OS.
`setclearance()`	None	In Trusted Extensions, the process clearance is the same as the label.
`setcmwlabel()`	`setflabel(3TSOL)`	Replaced with a library call.
`setcmwplabel()`	None	Files no longer have attributes.
`setegid()`, `seteuid()`	`setegid(2)`	Works as in Solaris OS.
`setfattrflag()`	None	Files no longer have flags.
`setfpriv()`	None	Files no longer have forced privileges.
`setgid()`	`setgid(2)`	Works as in Solaris OS.
`setgroups()`	`setgroups(2)`	Works as in Solaris OS.
`setpattr()`	`setpflags(2)`	Can set Trusted Extensions set of flags.

TABLE 5 System Call Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 System Call	Current Man Page	Note
setppriv()	setppriv(2)	Works as in Solaris OS.
setregid()	setregid(2)	Works as in Solaris OS.
setreuid()	setreuid(2)	Works as in Solaris OS.
setrlimit()	setrlimit(2)	Works as in Solaris OS.
setuid()	setuid(2)	Works as in Solaris OS.
shmat()	shmat(2)	Works as in Solaris OS.
shmctl()	shmctl(2)	Works as in Solaris OS.
shmdt()	shmdt(2)	Works as in Solaris OS.
shmget()	shmget(2)	Works as in Solaris OS.
shmgetl()	None	No longer needed.
shmop()	shmop(2)	Works as in Solaris OS.
sigsend(), sigsendset()	sigsend(2)	Works as in Solaris OS.
stat()	stat(2)	Works as in Solaris OS.
statvfs()	statvfs(2)	Works as in Solaris OS.
stime()	stime(2)	Works as in Solaris OS.
swapctl()	swapctl(2)	Works as in Solaris OS.
symlink()	symlink(2)	Works as in Solaris OS.
sysinfo()	sysinfo(2)	Works as in Solaris OS.
tokmapper()	None	TSIX functionality has been removed.
uadmin()	uadmin(2)	Works as in Solaris OS.
ulimit()	ulimit(2)	Works as in Solaris OS.
umount(), umount2()	umount(2)	Works as in Solaris OS.
unlink()	unlink(2)	Works as in Solaris OS.
utimes()	utimes(2)	Works as in Solaris OS.
vfork()	vfork(2)	Works as in Solaris OS.
write()	write(2)	Works as in Solaris OS.

TABLE 5 System Call Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 System Call	Current Man Page	Note
`writel()`	None	No longer needed.
`writev()`	`write(2)`	Works as in Solaris OS.
`writevl()`	None	No longer needed.

TABLE 6 Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris 8 Function	Current Man Page	Note
`au_preselect()`	`au_preselect(3BSM)`	Works as in Solaris OS.
`au_user_mask()`	`au_user_mask(3BSM)`	Works as in Solaris OS.
`endac()`	`getacinfo(3BSM)`	Works as in Solaris OS.
`endauclass()`	`getauclassent(3BSM)`	Works as in Solaris OS.
`endauevent()`	`getauevent(3BSM)`	Works as in Solaris OS.
`endauuser()`	`getauusernam(3BSM)`	Works as in Solaris OS.
`getacdir()`, `getacflg()`, `getacinfo()`, `getacmin()`, `getacna()`	`getacinfo(3BSM)`	Works as in Solaris OS.
`getauclassent()`, `getauclassent_r()`, `getauclassnam()`, `getauclassnam_r()`	`getauclassent(3BSM)`	Works as in Solaris OS.
`getauditflags()`, `getauditflagsbin()`, `getauditflagschar()`	`getauditflags(3BSM)`	Works as in Solaris OS.
`getauevent()`, `getauevent_r()`, `getauevnam()`, `getauevnam_r()`, `getauevnonam()`, `getauevnum()`, `getauevnum_r()`	`getauevent(3BSM)`	Works as in Solaris OS.
`getauuserent()`, `getauusernam()`	`getauusernam(3BSM)`	Works as in Solaris OS.
`getfauditflags()`	`getfauditflags(3BSM)`	Works as in Solaris OS.
`setac()`	`getacinfo(3BSM)`	Works as in Solaris OS.
`setauclass()`	`getauclassent(3BSM)`	Works as in Solaris OS.
`setauevent()`	`setauevent(3BSM)`	Works as in Solaris OS.

TABLE 6 Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Function	Current Man Page	Note
`setauuser()`	`getauusernam`(3BSM)	Works as in Solaris OS.
`endutent()`	`getutent`(3C)	Works as in Solaris OS.
`endutxent()`	`getutmp`(3C)	Works as in Solaris OS.
`ftw()`	`ftw`(3C)	Works as in Solaris OS.
`getutent()`, `getutid()`, `getutline()`	`getutent`(3C)	Works as in Solaris OS.
`getutmp()`, `getutmpx()`, `getutxent()` `getutxid()`, `getutxline()`	`getutmp`(3C)	Works as in Solaris OS.
`grantpt()`	`grantpt`(3C)	Works as in Solaris OS.
`initgroups()`	`initgroups`(3C)	Works as in Solaris OS.
`mlock()`	`mlock`(3C)	Works as in Solaris OS.
`mlockall()`	`mlockall`(3C)	Works as in Solaris OS.
`nftw()`	`nftw`(3C)	Works as in Solaris OS.
`plock()`	`plock`(3C)	Works as in Solaris OS.
`pututline()`	`getutent`(3C)	Works as in Solaris OS.
`pututxline()`	`getutmp`(3C)	Works as in Solaris OS.
`setutent()`	`setutent`(3C)	Works as in Solaris OS.
`setutxent()`	`setutxent`(3C)	Works as in Solaris OS.
`updwtmp()`, `updwtmpx()`	`getutmp`(3C)	Works as in Solaris OS.
`utmpname()`	`getutent`(3C)	Works as in Solaris OS.
`utmpxname()`	`getutmp`(3C)	Works as in Solaris OS.
`door_create()`	`door_create`(3DOOR)	Works as in Solaris OS.
`door_tcred()`	`door_ucred`(3DOOR)	Works as in Solaris OS.
`kstat_read()`, `kstat_write()`	`kstat_read`(3KSTAT)	Works as in Solaris OS.
`clnt_call()`	`rpc_clnt_calls`(3NSL)	Works as in Solaris OS.

TABLE 6 Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Function	Current Man Page	Note
`clnt_control()`, `clnt_create()`, `clnt_create_timed()`, `clnt_create_vers()`, `clnt_create_vers_timed()`, `clnt_destroy()`, `clnt_dg_create()`	`rpc_clnt_create`(3NSL)	Works as in Solaris OS.
`clnt_freeres()`, `clnt_geterr()`	`rpc_clnt_calls`(3NSL)	Works as in Solaris OS.
`clnt_pcreateerror()`	`rpc_clnt_create`(3NSL)	Works as in Solaris OS.
`clnt_perrno()`, `clnt_perror()`	`rpc_clnt_calls`(3NSL)	Works as in Solaris OS.
`clnt_raw_create()`, `clnt_spcreateerror()`	`rpc_clnt_create`(3NSL)	Works as in Solaris OS.
`clnt_sperrno()`, `clnt_sperror()`	`rpc_clnt_calls`(3NSL)	Works as in Solaris OS.
`clnt_tli_create()`, `clnt_tp_create()`, `clnt_tp_create_timed()`, `clnt_vc_create()`	`rpc_clnt_create`(3NSL)	Works as in Solaris OS.
`libt6()`	`ucred_get`(3C) `setsockopt`(3SOCKET)	`ucred_getlabel()` reads the socket label. `setsockopt()` with `SO_MAC_EXEMPT` option enables labeled communication.

TABLE 6 Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Function	Current Man Page	Note
NIS+ functions: `nis_add()`, `nis_add_entry()`, `nis_addmember()`, `nis_checkpoint()`, `nis_creategroup()`, `nis_destroygroup()`, `nis_first_entry()`, `nis_freeresult()`, `nis_freeservlist()`, `nis_freetags()`, `nis_getservlist()`, `nis_groups()`, `nis_ismember()`, `nis_list()`, `nis_lookup()` `nis_mkdir()` `nis_modify()`, `nis_modify_entry()`, `nis_names()`, `nis_next_entry()`, `nis_ping()`, `nis_print_group_entry()`, `nis_remove()`, `nis_remove_entry()`, `nis_removemember()`, `nis_rmdir()`, `nis_server()`, `nis_servstate()`, `nis_stats()`, `nis_tables()`, `nis_verifygroup()`	None	NIS+ is no longer a supported naming service.
`rpc()`	`rpc`(3NSL)	Works as in Solaris OS.
`rpc_broadcast()`, `rpc_broadcast_exp()`, `rpc_call()`, `rpc_clnt_calls()`	`rpc_clnt_calls`(3NSL)	Works as in Solaris OS.
`rpc_clnt_create()`, `rpc_createerr()`	`rpc_clnt_create`(3NSL)	Works as in Solaris OS.
`rpc_reg()`	`rpc_svc_reg`(3NSL)	Works as in Solaris OS.
`rpc_svc_calls()`	`rpc_svc_calls`(3NSL)	Works as in Solaris OS.
`rpc_svc_create()`	`rpc_svc_create`(3NSL)	Works as in Solaris OS.
`rpc_svc_reg()`	`rpc_svc_reg`(3NSL)	Works as in Solaris OS.
`rpcb_getaddr()`	`rpcbind`(3NSL)	Works as in Solaris OS.
`rpcb_getallmaps()`	None	
`rpcb_getmaps()`, `rpcb_gettime()`, `rpcb_rmtcall()`, `rpcb_set()`, `rpcb_unset()`, `rpcbind()`	`rpcbind`(3NSL)	Works as in Solaris OS.

TABLE 6 Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Function	Current Man Page	Note
`svc_auth_reg()`	`rpc_svc_reg`(3NSL)	Works as in Solaris OS.
`svc_control()`, `svc_create()`, `svc_destroy()`, `svc_dg_create()`	`rpc_svc_create`(3NSL)	Works as in Solaris OS.
`svc_dg_enablecache()`, `svc_done()`, `svc_exit()`	`rpc_svc_calls`(3NSL)	Works as in Solaris OS.
`svc_fd_create()`	`rpc_svc_create`(3NSL)	Works as in Solaris OS.
`svc_fdset()`, `svc_freeargs()`, `svc_getargs()`, `svc_getreq_common()`, `svc_getreq_poll()`, `svc_getreqset()`, `svc_getrpccaller()`, `svc_max_pollfd()`, `svc_pollfd()`	`rpc_svc_calls`(3NSL)	Works as in Solaris OS.
`svc_raw_create()`	`rpc_svc_create`(3NSL)	Works as in Solaris OS.
`svc_reg()`	`rpc_svc_reg`(3NSL)	Works as in Solaris OS.
`svc_run()`, `svc_sendreply()`	`rpc_svc_calls`(3NSL)	Works as in Solaris OS.
`svc_tli_create()`, `svc_tp_create()`	`rpc_svc_create`(3NSL)	Works as in Solaris OS.
`svc_unreg()`	`rpc_svc_reg`(3NSL)	Works as in Solaris OS.
`svc_vc_create()`	`rpc_svc_create`(3NSL)	Works as in Solaris OS.
T6 functions: `t6alloc_blk()`, `t6attr_query()`, `t6clear_blk()`, `t6cmp_blk()`, `t6copy_blk()`, `t6dup_blk()`, `t6ext_attr()`, `t6free_blk()`, `t6get_attr()`, `t6get_endpt_default()`, `t6get_endpt_mask()`, `t6last_attr()`, `t6new_attr()`, `t6peek_attr()`, `t6recvfrom()`, `t6sendto()`, `t6set_attr()`, `t6set_endpt_default()`, `t6set_endpt_mask()`, `t6size_attr()`	`ucred_get`(3C) `getsockopt`(3SOCKET) `setsockopt`(3SOCKET)	Replaced with `ucred_getlabel()`, `getsockopt()`, and `setsockopt()`.
`t_accept()`	`t_accept`(3NSL)	Works as in Solaris OS.
`t_bind()`	`t_bind`(3NSL)	Works as in Solaris OS.
`t_optmgmt()`	`t_optmgmt`(3NSL)	Works as in Solaris OS.

TABLE 6 Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Function	Current Man Page	Note
`t_snd()`	`t_snd`(3NSL)	Works as in Solaris OS.
`t_sndudata()`	`t_sndudata`(3NSL)	Works as in Solaris OS.
`xprt_register()`, `xprt_unregister()`	`rpc_svc_reg`(3NSL)	Works as in Solaris OS.
`dn_comp()`, `dn_expand()`, `fp_resstat()`, `herror()`, `hstrerror()`, `res_hostalias()`, `res_init()`, `res_mkquery()`, `res_nclose()`, `res_ninit()`, `res_nmkquery()`, `res_npquery()`, `res_nquery()`, `res_nquerydomain()`, `res_nsearch()`, `res_nsend()`, `res_nsendsigned()`, `res_query()`, `res_search()`, `res_send()`, `resolver()`	`resolver`(3RESOLV)	Works as in Solaris OS.
`clock_getres()`, `clock_gettime()`, `clock_settime()`	`clock_settime`(3RT)	Works as in Solaris OS.
`accept()`	`accept`(3SOCKET)	Works as in Solaris OS.
`bind()`	`bind`(3SOCKET)	Works as in Solaris OS.
`getsockopt()`	`getsockopt`(3SOCKET) and `getsockopt`(3XNET)	Adds `SO_MAC_EXEMPT` option.
`listen()`	`listen`(3SOCKET)	Works as in Solaris OS.
`send()`, `sendmsg()`, `sendto()`	`send`(3SOCKET)	Works as in Solaris OS.
`setsockopt()`	`setsockopt`(3SOCKET) and `setsockopt`(3XNET)	Adds `SO_MAC_EXEMPT` option.
`socket()`	`socket`(3SOCKET)	Works as in Solaris OS.

TABLE 7 TSOL Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris 8 Function	Current Man Page	Note
`Xbcleartos()`	`label_to_str`(3TSOL)	Obsolete.
`Xbcltos()`	None	CMW support removed.
`Xbsltos()`	`label_to_str`(3TSOL)	

TABLE 7 TSOL Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Function	Current Man Page	Note
`adornfc()`	None	Files are no longer adorned.
`auditwrite()`	None	Use the Solaris OS audit functions. This function is now private.
`aw_errno()`, `aw_geterrno()`, `aw_perror()`, `aw_perror_r()`, `aw_strerror()`	None	`auditwrite()` is now private. Use the au* equivalents.
`bclearhigh()`, `bclearlow()`	`str_to_label`(3TSOL)	Removed.
`bcleartoh()`, `bcleartoh_r()`, `bcleartos()`	`label_to_str`(3TSOL)	Obsolete. Label interfaces are now opaque.
`bclearundef()`, `bclearvalid()`	None	Removed.
`bclhigh()`, `bcllow()`	None	CMW support removed.
`bcltobanner()`	`label_to_str`(3TSOL)	Removed. Use new label translation functions.
`bcltoh()`, `bcltoh_r()`	None	CMW support removed.
`bcltos()`, `bcltosl()`	None	CMW support removed.
`bclundef()`	None	CMW support removed.
`blcompare()`, `bldominates()`, `blequal()`, `blinrange()`	`blcompare`(3TSOL)	No change.
`blinset()`, `blmanifest()`	None	Removed.
`blmaximum()`, `blminimum()`, `blminmax()`	`blminmax`(3TSOL)	No change.
`blportion()`	None	CMW support removed.
`blstrictdom()`	`blcompare`(3TSOL)	No change.
`bltocolor()`, `bltocolor_r()`	`label_to_str`(3TSOL)	Obsolete. Label interfaces are now opaque.
`bltype()`, `blvalid()`	None	Removed.
`bslhigh()`, `bsllow()`	`str_to_label`(3TSOL)	Replaced.
`bsltoh()`, `bsltoh_r()`, `bsltos()`	`label_to_str`(3TSOL)	Obsolete. Label interfaces are now opaque.
`bslundef()`, `bslvalid()`	None	Removed.

TABLE 7 TSOL Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Function	Current Man Page	Note
`btohex()`	`label_to_str`(3TSOL)	Obsolete. Label interfaces are now opaque.
`get_priv_text()`	`priv_gettext`(3C)	Replaced with Solaris function.
`getcsl()`	None	CMW support removed.
`getpeerinfo()`	`getpeerucred`(3C)	Replaced with Solaris function.
`getvfsaent()`, `getvfsafile()`	None	`vfstab_adjunct` file has been removed.
`h_alloc()`, `h_free()`	`label_to_str`(3TSOL)	Obsolete. Label interfaces are now opaque.
`hextob()`, `htobclear()`, `htobsl()`	`str_to_label`(3TSOL)	Obsolete. Label interfaces are now opaque.
`htobcl()`	None	CMW support removed.
`labelbuilder()`	`labelbuilder`(3TSOL)	IL and CMW modes removed.
`labelclipping()`	`labelclipping`(3TSOL)	IL and CMW modes removed.
`labelinfo()`		Removed.
`labelvers()`		Removed.
`mldgetcwd()`	None	No MLDs.
`mldlstat()`	None	No MLDs.
`mldrealpath()`	None	No MLDs.
`mldrealpathl()`	`getpathbylabel`(3TSOL)	Changed, and name changed to reflect new architecture.
`mldstat()`	None	No MLDs.
`priv_set_to_str()`, `priv_to_str()`	`priv_set_to_str`(3C)	Replaced with Solaris functions.
`randomword()`	None	No longer supported.
`sbcltos()`	None	CMW support removed.
`sbcleartos()`, `sbltos()`, `sbsltos()`	`str_to_label`(3TSOL)	Obsolete. Label interfaces are now opaque.
`set_effective_priv()`, `set_inheritable_priv()`, `set_permitted_priv()`	None	Use Solaris OS privilege interfaces.

TABLE 7 TSOL Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Function	Current Man Page	Note
`setbltype()`	None	Removed.
`setcsl()`, `stobcl()`	None	CMW support removed.
`stobclear()`, `stobl()`, `stobsl()`	`str_to_label`(3TSOL)	Obsolete. Label interfaces are now opaque.
`str_to_priv()`, `str_to_priv_set()`	None	Privileges are already strings.
`tsol_lbuild_create()`, `tsol_lbuild_destroy()`, `tsol_lbuild_get()`, `tsol_lbuild_set()`	`labelbuilder`(3TSOL)	IL and CMW modes removed.

The include file for the XTSOL library functions has been moved. Also, the man pages are now in the default `$MANPATH` directory.

TABLE 8 XTSOL Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris 8 Function	Current Man Page	Note
`XTSOLMakeTPWindow()`	`XTSOLMakeTPWindow`(3XTSOL)	See preceding paragraph for changes common to all functions.
`XTSOLShutdown()`	None	Removed.
`XTSOLgetClientAttributes()`	`XTSOLgetClientAttributes`(3XTSOL)	
`XTSOLgetPropAttributes()`	`XTSOLgetPropAttributes`(3XTSOL)	
`XTSOLgetPropLabel()`	`XTSOLgetPropLabel`(3XTSOL)	Acts on sensitivity label.
`XTSOLgetPropUID()`	`XTSOLgetPropUID`(3XTSOL)	
`XTSOLgetResAttributes()`	`XTSOLgetResAttributes`(3XTSOL)	
`XTSOLgetResLabel()`	`XTSOLgetResLabel`(3XTSOL)	Acts on sensitivity label.
`XTSOLgetResUID()`	`XTSOLgetResUID`(3XTSOL)	
`XTSOLgetWorkstationOwner()`	`XTSOLgetWorkstationOwner`(3XTSOL)	
`XTSOLsetPropLabel()`	`XTSOLsetPropLabel`(3XTSOL)	Acts on sensitivity label.
`XTSOLsetPropUID()`	`XTSOLsetPropUID`(3XTSOL)	
`XTSOLsetResLabel()`	`XTSOLsetResLabel`(3XTSOL)	Acts on sensitivity label.

TABLE 8 XTSOL Library Function Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Function	Current Man Page	Note
`XTSOLsetResUID()`	`XTSOLsetResUID`(3XTSOL)	
`XTSOLsetSessionHI()`	`XTSOLsetSessionHI`(3XTSOL)	
`XTSOLsetSessionLO()`	`XTSOLsetSessionLO`(3XTSOL)	
`XTSOLsetWorkstationOwner()`	`XTSOLsetWorkstationOwner`(3XTSOL)	
`XTSOLIsWindowTrusted()`	`XTSOLIsWindowTrusted`(3XTSOL)	
`XTSOLgetSSHeight()`	`XTSOLgetSSHeight`(3XTSOL)	
`XTSOLsetSSHeight()`	`XTSOLsetSSHeight`(3XTSOL)	
`XTSOLsetPolyInstInfo()`	`XTSOLsetPolyInstInfo`(3XTSOL)	

TABLE 9 Man Page Section 4 Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris 8 File	Current Man Page	Note
`audit.log`	`audit.log`(4)	Works as in Solaris OS.
`audit_class`	`audit_class`(4)	Trusted Extensions changes the X audit class masks.
`audit_control`	`audit_control`(4)	Works as in Solaris OS.
`audit_data`	`audit_data`(4)	Works as in Solaris OS.
`audit_event`	`audit_event`(4)	Works as in Solaris OS.
`audit_user`	`audit_user`(4)	Works as in Solaris OS.
`config.privs`	`TrustedExtensionsPolicy`(4)	Replaced.
`device_allocate`	None	The Device Allocation Manager is used to make changes to the device databases.
`device_maps`	`device_maps`(4)	Works as in Solaris OS.
`device_policy`	None	No longer needed.
`exec_attr`	`exec_attr`(4)	Style of entries are changed. Trusted Extensions uses the `solaris` policy keyword.
`fbtab`	`logindevperm`(4)	Works as in Solaris OS.
`inetd.conf`	`inetd.conf`(4)	Works as in Solaris OS.
`inittab`	`inittab`(4)	Works as in Solaris OS.

TABLE 9 Man Page Section 4 Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 File	Current Man Page	Note
label_encodings	label_encodings(4)	No change. Trusted Extensions installs a different default encodings file.
logindevperm	logindevperm(4)	Remains unsupported.
mnttab	mnttab(4)	Works as in Solaris OS.
nca.if	nca.if(4)	Remains unsupported.
nsswitch.conf	nsswitch.conf(4)	Works as in Solaris OS.
policy.conf	policy.conf(4)	Works as in Solaris OS.
priv_desc, priv_name	privileges(5)	Use privileges interface.
proc	proc(4)	Works as in Solaris OS.
resolv.conf	resolv.conf(4)	Works as in Solaris OS.
rmtab	rmtab(4)	Works as in Solaris OS.
sel_config	sel_config(4)	Replaced.
shadow	shadow(4)	Works as in Solaris OS.
sharetab	sharetab(4)	Works as in Solaris OS.
tndlog	None	Removed.
tnidb	None	Removed.
tnrhdb	tnrhdb(4)	Same format.
tnrhtp	tnrhtp(4)	Different format, two templates defined.
tsolgateways	None	Use the Solaris static routing mechanism.
tsolinfo	None	Special packaging utilities are no longer required.
user_attr	user_attr(4)	Trusted Extensions adds the idlecmd, idletime, clearance, labelview, and min_label keywords.
vfstab	vfstab(4)	Works as in Solaris OS.
vfstab_adjunct	None	File systems no longer have attributes.

TABLE 10 Man Page Section 5 Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris 8 Interface	Current Man Page	Note
`pam_tp_auth`	None	Removed.
`pam_tsol`	`pam_tsol_account(5)`	Trusted Extensions provides an authentication module only.
`pam_unix`	None	Replaced in Solaris OS. See `pam(3PAM)`.
`priv_macros`, `PRIV_ASSERT`, `PRIV_CLEAR`, `PRIV_EMPTY`, `PRIV_EQUAL`, `PRIV_FILL`, `PRIV_INTERSECT`, `PRIV_INVERSE`, `PRIV_ISASSERT`, `PRIV_ISEMPTY`, `PRIV_ISFULL`, `PRIV_ISSUBSET`, `PRIV_TEST`, `PRIV_UNION`, `PRIV_XOR`	None	Use the Solaris privilege interfaces, as described in "Privileges in Trusted Extensions" on page 19. See also "Privileges (Overview)" in *System Administration Guide: Security Services*.
`device_clean`	`device_clean(5)`	Has new options. Changed man page section.

TABLE 11 Device and Driver Interface Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases

Trusted Solaris 8 Device or Driver Interface	Current Man Page	Note
`sad()`	`sad(7D)`	Works as in Solaris OS.
`wscons()`	`wscons(7D)`	Works as in Solaris OS.
`kb()`	`kb(7M)`	Works as in Solaris OS.
`copyb()`	`copyb(9F)`	Works as in Solaris OS.
`copymsg()`	`copymsg(9F)`	Works as in Solaris OS.
`dupb()`	`dupb(9F)`	Works as in Solaris OS.
`dupmsg()`	`dupmsg(9F)`	Works as in Solaris OS.
`insq()`	`insq(9F)`	Works as in Solaris OS.
`kstat_create()`	`kstat_create(9F)`	Works as in Solaris OS.
`linkb()`	`linkb(9F)`	Works as in Solaris OS.
`msgpullup()`	`msgpullup(9F)`	Works as in Solaris OS.
`put()`	`put(9F)`	Works as in Solaris OS.
`putctl()`	`putctl(9F)`	Works as in Solaris OS.

TABLE 11 Device and Driver Interface Correspondences Between the Trusted Solaris 8 and Solaris Trusted Extensions Releases *(Continued)*

Trusted Solaris 8 Device or Driver Interface	Current Man Page	Note
`putctl1()`	`putctl1`(9F)	Works as in Solaris OS.
`putnext()`	`putnext`(9F)	Works as in Solaris OS.
`putnextctl()`	`putnextctl`(9F)	Works as in Solaris OS.
`putnextctl1()`	`putnextctl1`(9F)	Works as in Solaris OS.
`putq()`	`putq`(9F)	Works as in Solaris OS.
`tsol_get_strattr()`, `tsol_set_strattr()`	None	Removed.