



Man Pages (4): File Formats

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

Part No: 805-8070
November 1999

Copyright 1999 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, Trusted Solaris, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

Federal Acquisitions: Commercial Software – Government Users Subject to Standard License Terms and Conditions

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

Copyright 1999 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, Californie 94303-4900 Etats-Unis. Tous droits réservés.

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

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

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

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



Contents

PREFACE 5

Intro(4) 11

audit_class(4) 15

audit_control(4) 17

audit_data(4) 21

audit_event(4) 22

audit.log(4) 23

audit_user(4) 31

auth_desc(4) 32

auth_name(4) 41

config.privs(4) 43

device_allocate(4) 44

device_deallocate(4) 46

device_maps(4) 48

device_policy(4) 50

inetd.conf(4) 54

inittab(4) 57

label_encodings(4) 60

mnttab(4) 67

nsswitch.conf(4)	68
priv_desc(4)	75
priv_name(4)	88
resolv.conf(4)	90
rmtab(4)	94
sel_config(4)	95
sharetab(4)	97
tndlog(4)	98
tnidb(4)	99
tnrhdb(4)	103
tnrhtp(4)	105
tsolgateways(4)	121
tsolinfo(4)	125
tsolprof(4)	128
tsoluser(4)	132
vfstab(4)	135
vfstab_adjunct(4)	137
Index	142

PREFACE

Overview

A man page is provided for both the naive user and the sophisticated user who is familiar with the Trusted Solaris operating environment and is in need of online information. A man page is intended to answer concisely the question “What does it do?” The man pages in general comprise a reference manual. They are not intended to be a tutorial.

Trusted Solaris Reference Manual

In the AnswerBook2™ and online man command forms of the man pages, all man pages are available:

- Trusted Solaris man pages that are unique for the Trusted Solaris environment
- SunOS 5.7 man pages that have been changed in the Trusted Solaris environment
- SunOS 5.7 man pages that remain unchanged.

The printed manual, the *Trusted Solaris 7 Reference Manual* contains:

- Man pages that have been added to the SunOS operating system by the Trusted Solaris environment
- Man pages that originated in SunOS 5.7, but have been modified in the Trusted Solaris environment to handle security requirements.

Users of printed manuals need both manuals in order to have a full set of man pages, since the *SunOS5.7 Reference Manual* contains the common man pages that are not modified in the Trusted Solaris environment.

Man Page Sections

The following contains a brief description of each section in the man pages and the information it references:

- Section 1 describes, in alphabetical order, commands available with the operating system.
- Section 1M describes, in alphabetical order, commands that are used chiefly for system maintenance and administration purposes.
- Section 2 describes all of the system calls. Most of these calls have one or more error returns. An error condition is indicated by an otherwise impossible returned value.
- Section 3 describes functions found in various libraries, other than those functions that directly invoke UNIX system primitives, which are described in Section 2 of this volume.
- Section 4 outlines the formats of various files. The C structure declarations for the file formats are given where applicable.
- Section 5 contains miscellaneous documentation such as character set tables.
- Section 6 contains available games and demos.
- Section 7 describes various special files that refer to specific hardware peripherals, and device drivers. STREAMS software drivers, modules and the STREAMS-generic set of system calls are also described.
- Section 9 provides reference information needed to write device drivers in the kernel operating systems environment. It describes two device driver interface specifications: the Device Driver Interface (DDI) and the Driver/Kernel Interface (DKI).
- Section 9E describes the DDI/DKI, DDI-only, and DKI-only entry-point routines a developer may include in a device driver.
- Section 9F describes the kernel functions available for use by device drivers.
- Section 9S describes the data structures used by drivers to share information between the driver and the kernel.

Below is a generic format for man pages. The man pages of each manual section generally follow this order, but include only needed headings. For example, if there are no bugs to report, there is no BUGS section. See the `intro` pages for more information and detail about each section, and `man(1)` for more information about man pages in general.

NAME

This section gives the names of the commands or functions documented, followed by a brief description of what they do.

SYNOPSIS

This section shows the syntax of commands or functions. When a command or file does not exist in the standard path, its full pathname is shown. Options and arguments are alphabetized, with single letter arguments first, and options with arguments next, unless a different argument order is required.

The following special characters are used in this section:

- [] The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.
- . . . Ellipses. Several values may be provided for the previous argument, or the previous argument can be specified multiple times, for example, 'filename . . . '.
- | Separator. Only one of the arguments separated by this character can be specified at time.
- { } Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.

PROTOCOL

This section occurs only in subsection 3R to indicate the protocol description file.

DESCRIPTION

This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. It does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, functions and such, are described under USAGE.

IOCTL

This section appears on pages in Section 7 only. Only the device class which supplies appropriate parameters to the ioctl (2) system call is called `ioctl` and generates its own heading. `ioctl` calls for a specific device are listed alphabetically (on the man page for that specific device). `ioctl` calls are used for a particular class of devices all of which have an `io` ending, such as `mtio(7I)`

OPTIONS

This lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.

OPERANDS

This section lists the command operands and describes how they affect the actions of the command.

OUTPUT

This section describes the output - standard output, standard error, or output files - generated by the command.

RETURN VALUES

If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a function can return only constant values, such as 0 or -1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions declared void do not return values, so they are not discussed in RETURN VALUES.

ERRORS

On failure, most functions place an error code in the global variable `errno` indicating why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.

USAGE

This section is provided as a guidance on use. This section lists special rules, features and commands that require in-depth explanations. The subsections listed below are used to explain built-in functionality:

- Commands
- Modifiers
- Variables
- Expressions
- Input Grammar

EXAMPLES

This section provides examples of usage or of how to use a command or function. Wherever possible a complete example including command line entry and machine response is shown. Whenever an example is given, the prompt is shown as `example%` or if the user must be root, `example#`. Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS and USAGE sections.

ENVIRONMENT VARIABLES

This section lists any environment variables that the command or function affects, followed by a brief description of the effect.

EXIT STATUS

This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion and values other than zero for various error conditions.

FILES

This section lists all filenames referred to by the man page, files of interest, and files created or required by commands. Each is followed by a descriptive summary or explanation.

ATTRIBUTES

This section lists characteristics of commands, utilities, and device drivers by defining the attribute type and its corresponding value. See `attributes(5)` for more information.

SUMMARY OF TRUSTED SOLARIS CHANGES

This section describes changes to a Solaris 7 item by Trusted Solaris software. It is present in man pages that have been modified from Solaris 7 software.

SEE ALSO

This section lists references to other man pages, in-house documentation and outside publications. The references are divided into two sections, so that users of printed manuals can easily locate a man page in its appropriate printed manual.

DIAGNOSTICS

This section lists diagnostic messages with a brief explanation of the condition causing the error.

WARNINGS

This section lists warnings about special conditions which could seriously affect your working conditions. This is not a list of diagnostics.

NOTES

This section lists additional information that does not belong anywhere else on the page. It takes the form of an aside to the user, covering points of special interest. Critical information is never covered here.

BUGS

This section describes known bugs and wherever possible, suggests workarounds.

File Formats

NAME	Intro – Introduction to file formats
DESCRIPTION	<p>This section outlines the formats of various files. The C structure declarations for the file formats are given where applicable. Usually, the headers containing these structure declarations can be found in the directories <code>/usr/include</code> or <code>/usr/include/sys</code>. For inclusion in C language programs, however, the syntax <code>#include <filename.h></code> or <code>#include <sys/filename.h></code> should be used.</p> <p>Because the operating system now allows the existence of multiple file system types, there are several instances of multiple manual pages with the same name. These pages all display the name of the FSType to which they pertain, in the form <code>name_fstype</code> at the top of the page. For example, <code>fs_ufs(4)</code>.</p>
INTERFACES	<p>Descriptions of shared objects may include a definition of the global symbols that define the shared objects' public interface, for example <code>SUNW_1.1</code>. Other interfaces may exist within the shared object, for example <code>SUNW_private.1.1</code>. The public interface provides a stable, committed set of symbols for application development. The private interfaces are for internal use only, and may change at any time.</p> <p>For many shared objects, an archive library is provided for backward compatibility. Use of these libraries may restrict an applications ability to migrate between different Solaris releases. As dynamic linking is the preferred compilation method on Solaris, the use of these libraries is discouraged.</p>
TRUSTED SOLARIS DIFFERENCES	<p>In the Trusted Solaris environment, these configuration files can be:</p> <ul style="list-style-type: none"> ■ Files that are unique to and originate in the Trusted Solaris environment, such as <code>device_allocate(4)</code>. ■ SunOS 5.7 configuration files that have been modified to work within Trusted Solaris security policy, such as <code>proc(4)</code>. Man pages for modified files have been rewritten to remove information that is not accurate for how the file is used within the Trusted Solaris environment. Modified man pages also add descriptions for new fields or entities. ■ SunOS 5.7 files that remain unchanged from the Solaris 7 release, such as <code>timezone</code>. <hr/> <p>The printed <i>Trusted Solaris 7 Reference Manual</i> includes only those files that have been modified or originate in the Trusted Solaris environment. Printed versions of unchanged SunOS 5.7 man pages are found in the <i>SunOS 5.7 Reference Manual</i>. For more information on displaying manual pages, see Trusted Solaris Manual Page Display in <code>Intro(1)</code>.</p> <hr/> <p>The Trusted Solaris operating environment is a security-enhanced version of the Solaris operating environment, the Common Desktop Environment (CDE), the</p>

RULES FOR INCLUDING LABELS IN A CONFIGURATION FILE

X window system, and the Solstice AdminSuite set of system administration tools. To preserve security attributes, configuration files are usually not edited using `vi` or another common editor. Rather, administrative roles edit the files using the Solstice™ AdminSuite™ tools in the `Solstice_Apps` folder and the actions in the `System_Admin` folder in the Application Manager. These tools audit all changes and preserve the required owner, group, permissions and sensitivity labels of the files.

Follow the rules described here when entering labels in configuration files. When entering labels in graphical user interfaces, see *Rules for the Display and Entering of Labels in Intro(1)*. When entering labels on the command line in a UNIX shell, follow the rules in *Rules for the Display and Entering of Labels in Intro(1M)*.

Make sure that a program reading a configuration file can tell where the label starts and ends. Where the label is imbedded, as it is in the `device_allocate(4)` file, the only valid character to begin the label and terminate it is a semicolon (;). Most configuration files do not support label incrementations using plus or minus signs.

Configuration files are generally maintained at a sensitivity label of `ADMIN_LOW`. However, each site can choose whether to store labels in configuration files as text or as hexadecimal numbers, depending on the site's security policy, and the form used affects the sensitivity label at which the file should be stored. When labels are stored in human-readable form, the files that contain them must be protected at `ADMIN_HIGH`, so only administrative roles that have the `ADMIN_HIGH` label in their clearance can view the files. Also, if a file contains a collection of data written by all processes in the system (like the system log, `/dev/kmem`, and `/dev/mem` files) that file should be protected at the `ADMIN_HIGH` sensitivity label.

Labels entered in text form must be quoted.

POLICY FOR SECURITY ATTRIBUTES ON CONFIGURATION FILES

The default user and group for configuration files are `root` and `sys` and default permissions are `00644`. However, the security administrator should ensure that files that contain sensitivity information other than labels, such as those files that specify which activities are being audited, are not generally readable. These files should have more restrictive permissions, owner and group IDs, and possibly a protective label.

SEE ALSO

In Trusted Solaris, *Trusted Solaris Administrator's Procedures Trusted Solaris Developer's Guide*

Name	Description
<code>audit.log(4)</code>	Audit trail file

audit_class(4)	Audit class definitions
audit_control(4)	Control information for system audit daemon
audit_data(4)	Current information on audit daemon
audit_event(4)	Audit event definition and class mapping file
audit_user(4)	Per-user auditing data file
auth_desc(4)	Descriptions of defined authorizations
auth_name(4)	Authorization description database
config.privs(4)	List of window privileges that override system checks
device_allocate(4)	Device allocate in permissions
device_deallocate(4)	Device deallocate file
device_maps(4)	Maps device names to physical devices
device_policy(4)	Device policy file
inetd.conf(4)	Internet servers database
inittab(4)	Script for init
label_encodings(4)	Label encodings file
mnttab(4)	Mounted file system table
nsswitch.conf(4)	Configuration file for the name service switch
priv_desc(4)	Descriptions of defined privileges
priv_name(4)	Privilege description database
proc(4)	/proc, the process file system
resolv.conf(4)	Configuration file for name server routines
rmtab(4)	Remote mounted file system table
sel_config(4)	Selection rules for copy, cut, paste, drag and drop operations
sharetab(4)	Shared file system table
tndlog(4)	Log of tnd debugging information
tnidb(4)	Trusted network interface-control database
tnrhdb(4)	Trusted network remote-host database

<code>tnrntp(4)</code>	Trusted network remote-host templates
<code>tsolgateways(4)</code>	Static routing configuration file
<code>tsolinfo(4)</code>	Package security-attribute description file
<code>tsolprof(4)</code>	Trusted Solaris User Profiles Database
<code>tsoluser(4)</code>	Trusted Solaris User Security Attributes Database
<code>vfstab(4)</code>	Table of file system defaults
<code>vfstab_adjunct(4)</code>	Attribute data file for mounting a file system

NAME	audit_class – Audit class definitions
SYNOPSIS	/etc/security/audit_class
DESCRIPTION	<p>/etc/security/audit_class is a plain text system file that stores class definitions. Programs use the getauclassent(3) routines to access this information.</p> <p>The fields for each class entry are separated by colons. Each class entry is a bitmap and is separated from each other by a newline.</p> <p>Each entry in the audit_class file has the form:</p> <p><i>mask:name:description</i></p> <p>The fields are defined as follows:</p> <p><i>mask</i> The class mask.</p> <p><i>name</i> The class name.</p> <p><i>description</i> The description of the class.</p> <p>The classes are user-configurable. Each class is represented as a bit in the class mask which is an unsigned integer. Thus, there are 32 different classes available, plus two meta-classes, all and no.</p> <p>all represents a conjunction of all allowed classes, and is provided as a shorthand method of specifying all classes.</p> <p>no is the "invalid" class, and any event mapped solely to this class will not be audited. (Turning auditing on to the all meta class will <i>not</i> cause events mapped solely to the no class to be written to the audit trail.)</p>
EXAMPLES	<p>EXAMPLE 1 Here is a sample of an audit_class file:</p> <pre>0x00000000:no:invalid class 0x00000001:fr:file read 0x00000002:fw:file write 0x00000004:fa:file attribute access 0x00000008:fm:file attribute modify 0x00000010:fc:file create 0x00000020:fd:file delete 0x00000040:cl:file close 0xffffffff:all:all classes</pre>
SUMMARY OF TRUSTED SOLARIS CHANGES	By default, auditing is enabled in the Trusted Solaris environment. See <i>Trusted Solaris Audit Administration</i> for how to disable and enable auditing.
FILES	/etc/security/audit_class Audit class definitions.

SEE ALSO

**Trusted Solaris 7
Reference Manual**

getauclassent(3), audit_event(4)

NOTES

It is possible to deliberately turn on the `no` class in the kernel, in which case the audit trail will be flooded with records for the audit event `AUE_NULL`.

NAME	audit_control – Control information for system audit daemon
SYNOPSIS	/etc/security/audit_control
DESCRIPTION	<p>The audit_control file contains audit control information used by auditd(1M). Each line consists of a title and a string, separated by a colon. There are no restrictions on the order of lines in the file, although some lines must appear only once. A line beginning with # is a comment.</p> <p>Directory definition lines list the directories to be used when creating audit files, in the order in which they are to be used. The format of a directory line is:</p> <pre>dir: <i>directory-name</i></pre> <p><i>directory-name</i> is where the audit files will be created. Any valid writable directory can be specified.</p> <p>Unless explicitly told to look elsewhere, the auditreduce(1M) command by default looks for the audit trail in all directories named according to the following convention on the server on which the command is run. Therefore, this naming convention is recommended for directories in which audit-trail files are stored:</p> <pre>/etc/security/audit/<i>server</i>[.<i>number</i>]/files</pre> <p><i>server</i> is the name of the audit server on which the audit files are stored. The optional <i>number</i> is used when an audit server exports two or more audit partitions. For example, the audit server trustworthy exports /etc/security/audit/trustworthy and /etc/security/audit/trustworthy.1. For the current host to use both of these partitions, these lines must be added to the local audit_control file:</p> <pre>dir: /etc/security/audit/trustworthy/files dir: /etc/security/audit/trustworthy.1/files</pre> <p>Audit data may be stored in directories with other names at the discretion of the site. Some sites may want to store each host's audit data in a separate subdirectory. The audit structure used will depend on each individual site. If the defined audit structure differs from /etc/security/audit/*/files, auditreduce needs to be given the new location of the audit trail explicitly as described in auditreduce(1M).</p>

The audit threshold line specifies the percentage of free space that must be present in the file system containing the current audit file. The format of the threshold line is:

```
minfree: percentage
```

where *percentage* indicates the amount of free space required. If free space falls below this threshold, the audit daemon `auditd(1M)` invokes the shell script `audit_warn(1M)`. If no threshold is specified, the default is 0%.

The audit flags line specifies the default system audit value. This value is combined with the user audit value read from `audit_user(4)` to form the process audit state. The user audit value overrides the system audit value. The format of a flags line is:

```
flags: audit-flags
```

where *audit-flags* specifies which event classes are to be audited. The character string representation of *audit-flags* contains a series of flag names, each one identifying a single audit class, separated by commas. A name preceded by minus (-) means that the class should be audited for failure only; successful attempts are not audited. A name preceded by plus (+) means that the class should be audited for success only; failing attempts are not audited. Without a prefix, the name indicates that the class is to be audited for both successes and failures. The special string `all` indicates that all events should be audited: `-all` indicates that all failed attempts are to be audited; `+all`, all successful attempts. These prefixes turn off flags specified earlier in the string: caret (^), caret minus (^-), and caret plus (^+). Caret minus (^-) turns off audits for failed attempts; caret plus (^+) turns off audits for successful attempts; caret (^) turns off audits for both successful and failed attempts. These operators are typically used to reset flags.

The non-attributable flags line is similar to the flags line, but this one contains the audit flags that define what classes of events are audited when an action cannot be attributed to a specific user. The format of a `naflags` line is:

```
naflags: audit-flags
```

The flags are separated by commas, with no spaces.

The following table lists the predefined audit classes:

short name	long name	Short description
no	no_class	Null value for turning off event preselection
fr	file_read	Read of data, open for reading, etc.
fw	file_write	Write of data, open for writing, etc.

```

fa file_attr_acc Access of object attributes: stat, pathconf, etc.
fm file_attr_mod Change of object attributes: chown, flock, etc.
fc file_creation Creation of object
fd file_deletion Deletion of object
cl file_close close(2) system call
pc process process operations
nt network Network events: bind, connect, accept, etc.
ip ipc System V IPC operations
na non_attr Nonattributable events
ad administrative Administrative actions: mount, exportfs, etc.
lo login_logout Login and logout events
ap application Application auditing
io ioctl ioctl(2) system call
fn fcntl fcntl(2) system call
ot other Everything else
all all All flags set

```

Note that the classes are configurable; see `audit_class(4)`.

EXAMPLES

EXAMPLE 1 Sample `/etc/security/audit_control` file

Here is a sample `/etc/security/audit_control` file for the machine `eggplant`:

```

dir: /etc/security/jedgar/eggplant
dir: /etc/security/jedgar.aux/eggplant
#
# Last-ditch audit file system when jedgar fills up.
#
dir: /etc/security/global/eggplant
minfree: 20
flags: lo,ad,-all,^-fm
naflags: lo,ad

```

This identifies server `jedgar` with two file systems normally used for audit data, another server `global` used only when `jedgar` fills up or breaks, and specifies that the warning script is run when the file systems are 80% filled. It also specifies that all logins, administrative operations are to be audited (whether or not they succeed), and that failures of all types except failures to access object attributes are to be audited.

SUMMARY OF TRUSTED SOLARIS CHANGES

By default, the machine halts when audit files run out of disk space. The Trusted Solaris environment adds programming interfaces, audit tokens, audit classes, and audit events.

By default, auditing is enabled in the Trusted Solaris environment. See *Trusted Solaris Audit Administration* for how to disable and enable auditing.

SEE ALSO

**Trusted Solaris 7
Reference Manual**

audit(1M), audit_warn(1M), auditd(1M), audit(2), getfauditflags(3),
audit.log(4), audit_class(4), audit_user(4), *Trusted Solaris Audit
Administration*

NAME	audit_data – Current information on audit daemon		
SYNOPSIS	/etc/security/audit_data		
DESCRIPTION	<p>The audit_data file contains information about the audit daemon. The file contains the process ID of the audit daemon, and the pathname of the current audit log file. The format of the file is:</p> <p><i>pid: pathname</i></p> <p>Where <i>pid</i> is the process ID for the audit daemon, and <i>pathname</i> is the full pathname for the current audit log file.</p>		
EXAMPLES	<p>EXAMPLE 1 A sample audit_data file.</p> <pre>64:/etc/security/audit/iedgar/19990506081249.19990506230945.eggplant</pre>		
SUMMARY OF TRUSTED SOLARIS CHANGES	<p>By default, auditing is enabled in the Trusted Solaris environment. The audit_data file is protected at ADMIN_HIGH.</p> <p>See <i>Trusted Solaris Audit Administration</i> for how to disable and enable auditing.</p>		
FILES	<table> <tr> <td>/etc/security/audit_data</td><td>Current information on audit daemon.</td></tr> </table>	/etc/security/audit_data	Current information on audit daemon.
/etc/security/audit_data	Current information on audit daemon.		
SEE ALSO	audit(1M), auditd(1M), audit(2), audit.log(4)		
Trusted Solaris 7 Reference Manual	<i>Trusted Solaris Audit Administration</i>		

NAME	audit_event – Audit event definition and class mapping file	
SYNOPSIS	/etc/security/audit_event	
DESCRIPTION	<p>/etc/security/audit_event is a plain text system file that stores event definitions and specifies the event-to-class mappings. Programs use the getauevent(3) routines to access this information.</p> <p>The fields for each event entry are separated by colons. Each event is separated from the next by a newline.</p> <p>Each entry in the audit_event file has the form:</p> <p><i>number:name:description: flags</i></p> <p>The fields are defined as follows:</p> <p><i>number</i> The event number.</p> <p><i>name</i> The event name.</p> <p><i>description</i> The description of the event.</p> <p><i>flags</i> Flags specifying classes to which the event is mapped.</p>	
EXAMPLES	<p>EXAMPLE 1 Some audit_event file entries</p> <pre> 7:AUE_EXEC:exec(2):ps 79:AUE_OPEN_WTC:open(2) - write,creat,trunc:fc,fd,fw 6152:AUE_login:login - local:lo 6153:AUE_logout:logout:lo 6154:AUE_telnet:login - telnet:lo 6155:AUE_rlogin:login - rlogin:lo </pre>	
SUMMARY OF TRUSTED SOLARIS CHANGES	<p>Programs use the getauevent(3) routines rather than the getauevent(3) routines to access this information.</p> <p>By default, auditing is enabled in the Trusted Solaris environment. See <i>Trusted Solaris Audit Administration</i> for how to disable and enable auditing.</p>	
FILES	/etc/security/audit_event	Audit event definition and class mapping file.
SEE ALSO	getauevent(3), audit_control(4)	
Trusted Solaris 7 Reference Manual	<i>Trusted Solaris Audit Administration</i>	

NAME	audit.log – Audit trail file																				
SYNOPSIS	<pre>#include <bsm/audit.h> #include <bsm/audit_record.h></pre>																				
DESCRIPTION	<p>audit.log files are the depository for audit records stored locally or on an audit server. These files are kept in directories named in the file audit_control(4). They are named to reflect the time they are created and are, when possible, renamed to reflect the time they are closed as well. The name takes the form</p> <pre>yyyymmddhhmmss.not_terminated.hostname</pre> <p>when open or if the auditd(1M) terminated ungracefully, and the form</p> <pre>yyyymmddhhmmss.yyyyymmddhhmmss.hostname</pre> <p>when properly closed. <i>yyyy</i> is the year, <i>mm</i> the month, <i>dd</i> day in the month, <i>hh</i> hour in the day, <i>mm</i> minute in the hour, and <i>ss</i> second in the minute. All fields are of fixed width.</p> <p>The audit.log file begins with a standalone <i>file token</i> and typically ends with one also. The beginning file token records the pathname of the previous audit file, while the ending file token records the pathname of the next audit file. If the file name is NULL the appropriate path was unavailable.</p> <p>The audit.log files contains audit records. Each audit record is made up of <i>audit tokens</i>. Each record contains a header token followed by various data tokens. Depending on the audit policy in place by auditon(2), optional other tokens such as trailers or sequences may be included.</p> <p>The tokens are defined as follows:</p> <p>The <i>file</i> token consists of:</p> <table> <tr> <td>token ID</td><td>char</td></tr> <tr> <td>seconds of time</td><td>uint_t</td></tr> <tr> <td>milliseconds of time</td><td>uint_t</td></tr> <tr> <td>file name length</td><td>short</td></tr> <tr> <td>file pathname</td><td>null terminated string</td></tr> </table> <p>The <i>header</i> token consists of:</p> <table> <tr> <td>token ID</td><td>char</td></tr> <tr> <td>record byte count</td><td>ulong_t</td></tr> <tr> <td>version #</td><td>char (1)</td></tr> <tr> <td>event type</td><td>ushort_t</td></tr> <tr> <td>event modifier</td><td>ushort_t</td></tr> </table>	token ID	char	seconds of time	uint_t	milliseconds of time	uint_t	file name length	short	file pathname	null terminated string	token ID	char	record byte count	ulong_t	version #	char (1)	event type	ushort_t	event modifier	ushort_t
token ID	char																				
seconds of time	uint_t																				
milliseconds of time	uint_t																				
file name length	short																				
file pathname	null terminated string																				
token ID	char																				
record byte count	ulong_t																				
version #	char (1)																				
event type	ushort_t																				
event modifier	ushort_t																				

seconds of time	uint_t
milliseconds of time	uint_t

The *trailer* token consists of:

token ID	char
trailer magic number	ushort_t
record byte count	ulong_t

The *arbitrary data* token is defined:

token ID	char
how to print	char
basic unit	char
unit count	char
data items	

depends on basic unit

The *in_addr* token consists of:

token ID	char
internet address	char

The *ip* token consists of:

token ID	char
version and ihl	char
type of service	char
length	short
id	ushort_t
offset	ushort_t
ttl	char
protocol	char
checksum	ushort_t
source address	long
destination address	long

The *ipport* token consists of:

token ID	char
port address	short

The *opaque* token consists of:

token ID	char
size	short
data	char, <i>size</i> chars

The *path* token consists of:

token ID	char
path length	short
path	null terminated string

The *process* token consists of:

token ID	char
auid	ulong_t
euid	ulong_t
egid	ulong_t
ruid	ulong_t
rgid	ulong_t
pid	ulong_t
sid	ulong_t
terminal ID	ulong_t (port ID)
ulong_t	(machine ID)

The *return* token consists of:

token ID	char
error number	char
return value	long

The *subject* token consists of:

token ID	char
auid	ulong_t
euid	ulong_t
egid	ulong_t
ruid	ulong_t
rgid	ulong_t
pid	ulong_t
sid	ulong_t
terminal ID	ulong_t (port ID)
ulong_t	(machine ID)

The *System V IPC* token consists of:

token ID	char
object ID type	char
object ID	long

The *text* token consists of:

token ID	char
text length	short
text	null terminated string

The *attribute* token consists of:

token ID	char
mode	ulong_t
uid	ulong_t
gid	ulong_t
file system id	long
node id	long
device	ulong_t

The *groups* token consists of:

token ID	char
number	short
group list	long, <i>size</i> chars

The *System V IPC permission* token consists of:

token ID	char
uid	ulong_t
gid	ulong_t
cuid	ulong_t
cgid	ulong_t
mode	ulong_t
seq	ulong_t
key	long

The *arg* token consists of:

token ID	char
argument #	char
argument value	long
string length	short
text	null terminated string

The *exec_args* token consists of:

token ID	char
count	long
text	<i>count</i> null terminated string(s)

The *exec_env* token consists of:

token ID	char
count	long
text	<i>count</i> null terminated string(s)

The *exit* token consists of:

token ID	char
status	long
return value	long

The *socket* token consists of:

token ID	char
socket type	short
local port	short
local Internet address	char
remote port	short
remote Internet address	char

The *seq* token consists of:

token ID	char
sequence number	long

The *acl* token consists of

token ID	char
num of entries	int
(following three fields repeated num times)	
object type	int
uid/gid	int
permissions	short

The *clearance* token consists of

token ID	char
CLEARANCE	
label ID	char
pad character	char
classification	short
compartments	8 ints

The *host* token consists of

token ID	char
local Internet address	long

The *liaison* token consists of

token ID	char
liaison ID	int

The *priv* token consists of

token ID	char
succ/fail	char
priv. used	int

The *privilege* token consists of

token ID	char
type of set	char
priv. set	4 ints

The *slabel* token consists of

token ID	char
SLABEL	
pad character	char
classification	short
compartments	8 ints

The *xatom* token consists of

token ID	char
string length	short
atom string	string length bytes

The *xcolormap* token consists of

token ID	char
XID	int
creator UID	int

The *xcursor* token consists of

token ID	char
XID	int
creator UID	int

The *xfont* token consists of

token ID	char
XID	int
creator UID	int

The *xgc* token consists of

token ID	char
XID	int
creator UID	int

The *xpixmap* token consists of

token ID	char
XID	int
creator UID	int

The *xproperty* token consists of

token ID	char
XID	int
creator UID	int
string length	short
string	string length bytes

The *xselect* token consists of

token ID	char
property length	short
property string	property length bytes
prop. type len.	short
prop type	prop. type len. bytes
data length	short
window data	data length bytes

The *xwindow* token consists of

XID	int
creator UID	int

SUMMARY OF TRUSTED SOLARIS CHANGES

These audit tokens have been added to the Trusted Solaris auditing module: `acl`, `clearance`, `host`, `liaison`, `priv`, `privilege`, `slabel`, `xatom`, `xcolormap`, `xcursor`, `xfont`, `xgc`, `xpixmap`, `xproperty`, `xselect`, and `xwindow`. Trusted Solaris auditing also uses `auditwrite(3)` instead of `au_to_*()` function calls to create audit tokens.

Information labels (ILs) are not supported in Trusted Solaris 7 and later releases. Trusted Solaris software interprets any ILs on communications and files from systems running earlier releases as `ADMIN_LOW`.

Objects still have CMW labels, and CMW labels still include the IL component: `IL[SL]`; however, the IL component is fixed at `ADMIN_LOW`.

As a result, Trusted Solaris 7 has the following characteristics:

- ILs do not display in window labels; SLs (Sensitivity Labels) display alone within brackets.
- ILs do not float.
- Setting an IL on an object has no effect.
- Getting an object's IL will always return `ADMIN_LOW`.
- Although certain utilities, library functions, and system calls can manipulate IL strings, the resulting ILs are always `ADMIN_LOW`, and cannot be set on any objects.
- Options related to information labels in the `label_encodings(4)` file can be ignored:

```
Markings Name= Marks;
Float Process Information Label;
```

SEE ALSO

Trusted Solaris 7
Reference Manual

SunOS 5.7 Reference
Manual

`audit(1M)`, `auditd(1M)`, `audit(2)`, `audition(2)`, `auditwrite(3)`,
`audit_control(4)`

`au_to(3)`

NAME	audit_user – Per-user auditing data file							
SYNOPSIS	/etc/security/audit_user							
DESCRIPTION	<p>audit_user is an access-restricted plain text system file that stores per-user auditing preselection data. Programs use the getauusernam(3) to access this information.</p> <p>The fields for each user entry are separated by colons. Each user is separated from the next by a newline. audit_user does not have general read permission.</p> <p>Each entry in the audit_user file has the form:</p> <pre>username:always-audit-flags:never-audit-flags</pre> <p>The fields are defined as follows:</p> <table><tr><td>username</td><td>The user's login name.</td></tr><tr><td>always-audit-flags</td><td>Flags specifying event classes to always audit.</td></tr><tr><td>never-audit-flags</td><td>Flags specifying event classes to never audit.</td></tr></table>		username	The user's login name.	always-audit-flags	Flags specifying event classes to always audit.	never-audit-flags	Flags specifying event classes to never audit.
username	The user's login name.							
always-audit-flags	Flags specifying event classes to always audit.							
never-audit-flags	Flags specifying event classes to never audit.							
EXAMPLES	<p>EXAMPLE 1 Sample audit_user file.</p> <p>Here is a sample audit_user file:</p> <pre>other:lo,ad:io,cl freda:lo,ex,+fc,-fr,-fa:io,cl ethel:lo,ex,nt:io,cl</pre>							
SUMMARY OF TRUSTED SOLARIS CHANGES	<p>By default, auditing is enabled in the Trusted Solaris environment. See <i>Trusted Solaris Audit Administration</i> for how to disable and enable auditing.</p>							
FILES	/etc/security/audit_user	Per-user auditing data file.						
	/etc/passwd	Per-machine user password file.						
SEE ALSO	getauusernam(3), audit_control(4)							
Trusted Solaris 7 Reference Manual	Trusted Solaris Audit Administration							
SunOS 5.7 Reference Manual	passwd(4)							

NAME	auth_desc – Descriptions of defined authorizations
SYNOPSIS	<code>#include <tsol/auth.h></code>
DESCRIPTION	<p>Every defined authorization has a manifest constant for use in programs, a name for use in user interfaces, and a description displayed by certain administrative tools. A set of authorizations is assigned to an execution profile, and the execution profile is assigned to a user or role. The user or role to which the execution profile is assigned has the authority to perform the tasks allowed by the authorizations in the execution profile. Authorizations are interpreted by utility and desktop tools. The manifest constant, name, and description for each authorization defined on this system follow.</p> <p>Manifest Constant <code>TSOL_AUTH_ENABLE_LOGIN</code></p> <p>Name enable logins</p> <p> Allows a user to enable logins on a machine that was just booted. Until logins are enabled there is no interactive use of the machine's resources.</p> <p>Manifest Constant <code>TSOL_AUTH_REMOTE_LOGIN</code></p> <p>Name remote login</p> <p> Allows a user to remotely login, using programs such as TELNET, or FTP, in a way that requires entering identification and authentication information. Such a login is different from extending an existing login from one machine to another without re-authentication because the trusted path is not guaranteed for these methods.</p> <p>Manifest Constant <code>TSOL_AUTH_TERMINAL_LOGIN</code></p> <p>Name terminal login</p> <p> Allows a user to login via a serial port. Such a login is different from extending an existing login from one machine to another without re-authentication because there is no trusted path presented for entering either the identification or authentication information.</p> <p>Manifest Constant <code>TSOL_AUTH_FILE_AUDIT</code></p> <p>Name set/get file audit flags</p> <p> Allows a user to specify or view the auditing preselection information or public object flag to be associated with access to a file or directory. The auditing preselection information may override the preselection information associated with a user's access to a file or directory. The public object flag may override the successful read/search access</p>

preselection information associated with a user's access to a file or directory.

Manifest Constant TSOL_AUTH_FILE_DOWNGRADE_SL

Name downgrade file sensitivity label

Allows a user to specify the Sensitivity Label to set on a file that does not dominate the file's existing Sensitivity Label.

Manifest Constant TSOL_AUTH_FILE_UPGRADE_SL

Name upgrade file sensitivity label

Allows a user to specify the Sensitivity Label to set on a file that dominates the file's existing Sensitivity Label.

Manifest Constant TSOL_AUTH_FILE_OWNER

Name act as file owner

Allows a user to act as a file's owner. This includes the ability to change the permission bits and ACL, to downgrade the Sensitivity Label, and set privileges (if further authorized by set file privileges) of files not owned. Also included is the ability to read and search directories, copy, move, and delete files not owned.

Manifest Constant TSOL_AUTH_FILE_CHOWN

Name change file owner

Allows a user to change the ownership and group of a file.

Manifest Constant TSOL_AUTH_FILE_SETPRIV

Name set file privileges

Allows a user to specify the allowed and forced privileges to be associated with a the execution of a program file.

Manifest Constant TSOL_AUTH_ALLOCATE

Name allocate device

Allows a user to allocate a device and specify the CMW label to associate with information imported from it, or exported to it.

Manifest Constant TSOL_AUTH_CONFIG_DEVICE

Name configure device attributes

Allows an administrator to configure a device. Device configuration includes such things as setting the device name, type, label range, allocatable status, and allocation authorization list.

Manifest Constant	TSOL_AUTH_REVOKE_DEVICE
Name	revoke or reclaim device
	Allows an administrator to deallocate a currently allocated device or reset the allocate error state to make a device allocatable again.
Manifest Constant	TSOL_AUTH_WIN_DOWNGRADE_SL
Name	paste to a downgraded window
	Allows a user to paste selected information to a window whose Sensitivity Label does not dominate the selected information's Sensitivity Label.
Manifest Constant	TSOL_AUTH_WIN_UPGRADE_SL
Name	paste to an upgraded window
	Allows a user to paste selected information to a window whose Sensitivity Label dominates the selected information's Sensitivity Label.
Manifest Constant	TSOL_AUTH_SYS_ACCRED_SET
Name	use all defined labels
	Allows a user to use all the available labels on the system rather than to be restricted to just the labels in the label encodings defined user accreditation range. Using a label implies the ability to specify that label for any of the label building interfaces which include those to re-label files and create workspaces.
Manifest Constant	TSOL_AUTH_BYPASS_FILE_VIEW
Name	bypass file view
	Allows a user to drag and drop a file without viewing that file's contents.
Manifest Constant	TSOL_AUTH_SHUTDOWN
Name	shut down the system
	Allows a user to shut down the system via the Trusted Path menu. When the system is a CDE X Terminal, the CDE X Terminal is shut down, not the server. Unless the "abort_enable" system variable (see <code>/etc/system</code>) is set to 0, this authorization can be bypassed by entering the keyboard abort sequence, gaining entry to the PROM, and rebooting from the PROM.
Manifest Constant	TSOL_AUTH_USER_INDENT

Name	set user identity	
	Allows an administrator to set the security information related to the user's identity. The user name, primary group, secondary groups, comment, and login shell may all be set via the User Manager. This authorization is needed to add, copy, or delete a user.	
Manifest Constant		TSOL_AUTH_USER_PASSWORD
Name	set user password	
	Allows an administrator to set password information pertaining to a user. The password, type of password, life time, expiration date, warning days, and the permission to set up the credentials table may all be set via the User Manager.	
Manifest Constant		TSOL_AUTH_USER_SELF
Name	permit self-modification	
	Allows an administrator to modify his or her own user attributes.	
Manifest Constant		TSOL_AUTH_USER_LABELS
Name	set user labels	
	Allows an administrator to set various label-related pieces of information associated with a particular user. A user's minimum login label, clearance, label view, and label translation attributes may be set via the User Manager.	
Manifest Constant		TSOL_AUTH_USER_AUDIT
Name	set user audit flags	
	Allows an administrator to set the per user audit flags.	
Manifest Constant		TSOL_AUTH_USER_PROFILES
Name	set user profiles	
	Allows an administrator to assign profiles to a user.	
Manifest Constant		TSOL_AUTH_USER_IDLE
Name	set idle time	
	Allows an administrator to set the idle time and determine which action to take when a workstation has been idle for too long. The idle time and idle command can be set via the User Manager.	
Manifest Constant		TSOL_AUTH_USER_ROLES
Name	set roles list	

	Allows an administrator to select which roles a user may assume. When a user assumes a role he or she may use all commands and actions granted to that role.
Manifest Constant	TSOL_AUTH_USER_HOME
Name	set home directory attributes
	Allows an administrator to determine such things as location, permissions, and initial contents of a user's home directory.
Manifest Constant	TSOL_AUTH_PRINT_ADMIN
Name	administer printing
	Allows a user to perform Trusted Printing System administration. Allows a user to start and stop printing daemons, list and cancel other users' print jobs, etc.
Manifest Constant	TSOL_AUTH_PRINT_CANCEL
Name	cancel any print job
	Allows user to cancel a print request queued by any user.
Manifest Constant	TSOL_AUTH_PRINT_LIST
Name	list all print jobs
	Allows a user to get a list of queued print jobs for all users.
Manifest Constant	TSOL_AUTH_PRINT_MAC_OVERRIDE
Name	bypass print system mac check
	Allows a user to cancel or list print jobs at any sensitivity label.
Manifest Constant	TSOL_AUTH_PRINT_NOBANNER
Name	print without banners
	Allows a user to submit a print request to the Trusted Printing System that specifies (by means of the 'lp -o nobanner' option) that the print job's banner and trailer pages should be suppressed.
Manifest Constant	TSOL_AUTH_PRINT_POSTSCRIPT
Name	print a PostScript file
	Allows a user to print a PostScript file with the Trusted Printing System.
Manifest Constant	TSOL_AUTH_PRINT_UNLABELED
Name	print without labels

Allows a user to submit a print request to the Trusted Printing System that specifies (by means of the 'lp -o nolabels' option) that the body pages of the print job should have the top and bottom labels suppressed.

Manifest Constant TSOL_AUTH_DB_ALIASES

Name modify aliases

Allows a user to edit the aliases databases via the Database Manager.

Manifest Constant TSOL_AUTH_DB_AUTO_HOME

Name modify auto_home

Allows a user to edit the auto_home databases via the Database Manager.

Manifest Constant TSOL_AUTH_DB_BOOTPARAMS

Name modify bootparams

Allows a user to edit the bootparams databases via the Database Manager.

Manifest Constant TSOL_AUTH_DB_ETHERS

Name modify ethers

Allows a user to edit the ethers databases via the Database Manager.

Manifest Constant TSOL_AUTH_DB_HOSTS

Name modify hosts

Allows a user to edit the hosts databases via the Database Manager.

Manifest Constant TSOL_AUTH_DB_LOCALE

Name modify locale

Allows a user to edit the locale databases via the Database Manager.

Manifest Constant TSOL_AUTH_DB_NETGROUP

Name modify netgroup

Allows a user to edit the netgroup databases via the Database Manager.

Manifest Constant TSOL_AUTH_DB_NETMASKS

Name modify netmasks

	Allows a user to edit the netmasks databases via the Database Manager.
Manifest Constant	TSOL_AUTH_DB_NETWORKS
Name	modify networks
	Allows a user to edit the networks databases via the Database Manager.
Manifest Constant	TSOL_AUTH_DB_PASSWD
Name	modify password
	Allows a user to edit the password databases via the Database Manager.
Manifest Constant	TSOL_AUTH_DB_PROTOCOLS
Name	modify protocols
	Allows a user to edit the protocols databases via the Database Manager.
Manifest Constant	TSOL_AUTH_DB_RPC
Name	modify rpc
	Allows a user to edit the rpc databases via the Database Manager.
Manifest Constant	TSOL_AUTH_DB_SERVICES
Name	modify services
	Allows a user to edit the services databases via the Database Manager.
Manifest Constant	TSOL_AUTH_DB_TIMEZONE
Name	modify timezone
	Allows a user to edit the timezone databases via the Database Manager.
Manifest Constant	TSOL_AUTH_DB_TNIDB
Name	modify tnidb
	Allows a user to edit the tnidb databases via the Database Manager.
Manifest Constant	TSOL_AUTH_DB_TNRHDB
Name	modify tnrhdb
	Allows a user to edit the tnrhdb databases via the Database Manager.
Manifest Constant	TSOL_AUTH_DB_TNRHTP

Name modify tnhttp
Allows a user to edit the tnhttp databases via the Database Manager.

Manifest Constant TSOL_AUTH_CRON_ADMIN

Name modify cron admin
Allows a user to modify and list crontab files of role users and users named in `/etc/cron.d/cron.admin`.

Manifest Constant TSOL_AUTH_CRON_USER

Name modify cron users
Allows a user to modify and list crontab files of non-administrative users.

Manifest Constant TSOL_AUTH_AT_ADMIN

Name modify at admin
Allows a user to remove and list at jobs of role users and users named in `/etc/cron.d/at.admin`.

Manifest Constant TSOL_AUTH_AT_USER

Name modify at users
Allows a user to remove and list all jobs of non-administrative users.

FILES

`/usr/lib/tsol/locale/locale/auth_name`
Authorizations descriptions

`</usr/include/tsol/auth_names.h>`
Manifest constant and ID value definitions

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsu

NOTES

Information labels (ILs) are not supported in Trusted Solaris 7 and later releases. Trusted Solaris software interprets any ILs on communications and files from systems running earlier releases as `ADMIN_LOW`.

Objects still have CMW labels, and CMW labels still include the IL component: `IL[SL]`; however, the IL component is fixed at `ADMIN_LOW`.

As a result, Trusted Solaris 7 has the following characteristics:

- ILs do not display in window labels; SLs (Sensitivity Labels) display alone within brackets.
- ILs do not float.
- Setting an IL on an object has no effect.
- Getting an object's IL will always return ADMIN_LOW.
- Although certain utilities, library functions, and system calls can manipulate IL strings, the resulting ILs are always ADMIN_LOW, and cannot be set on any objects.
- Options related to information labels in the label_encodings(4) file can be ignored:

```
Markings Name= Marks;  
Float Process Information Label;
```

SEE ALSO

**Trusted Solaris 7
Reference Manual**

Intro(3), auth_to_str(3), chkauth(3), auth_name(4)

Trusted Solaris administrator's document set, Trusted Solaris Developer's Guide

**SunOS 5.7 Reference
Manual**

attributes(5)

NAME	auth_name – Authorization description database				
SYNOPSIS	/usr/lib/tsxol/locale/ <i>locale</i> /auth_name				
DESCRIPTION	<p>The auth_name database defines the localized authorization names and descriptions defined on this system. This database is used along with <tsxol/auth_names.h> by auth_to_str(3), str_to_auth(3), and get_auth_text(3) to translate between authorization ID, authorization name string, and description.</p> <p>Each entry in the auth_name database consists of one line with fields separated by colons (:). A line ending with a backslash (\) indicates continuation of the entry on the next line. Lines beginning with a # character are treated as comments. Each entry has the form:</p> <p><i>constant</i> : <i>name</i> : <i>description</i></p> <p>The entry fields are:</p> <p><i>constant</i> The <i>constant</i> field must be identical to the manifest constant defined for the authorization in <tsxol/auth_names.h>, where a unique authorization ID is assigned to each authorization constant.</p> <p><i>name</i> The external name of the authorization. It is returned by auth_to_str() and is used by str_to_auth(). Authorization names are concise and descriptive so that they may be used in various GUIs. The authorization name can be customized and localized.</p> <p><i>description</i> The description of the activity permitted by the authorization. The description name can be customized and localized.</p>				
ATTRIBUTES	<p>See attributes(5) for descriptions of the following attributes:</p> <table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr> </thead> <tbody> <tr> <td>Availability</td><td>SUNWtsu</td></tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWtsu
ATTRIBUTE TYPE	ATTRIBUTE VALUE				
Availability	SUNWtsu				
EXAMPLES	<p>EXAMPLE 1 Two auth_name entries</p> <pre># # Example entries in /usr/lib/tsxol/locale/C/auth_name # TSOL_AUTH_ENABLE_LOGIN:enable logins:Allows a user to enable logins on a \ machine that was just booted. Until logins are enabled there is \ no interactive use of the machine's resources. TSOL_AUTH_REMOTE_LOGIN:remote login:Allows a user to remotely login, using \ programs such as TELNET, or FTP, in a way that requires entering \ identification and authentication information. Such a login is \ different from extending an existing login from one machine to \</pre>				

another without re-authentication because the trusted path is not \ guaranteed for these methods.

SEE ALSO

**Trusted Solaris 7
Reference Manual**

**SunOS 5.7 Reference
Manual**

auth_to_str(3), chkauth(3), auth_desc(4)

attributes(5)

NAME	config.privs – List of window privileges that override system checks				
SYNOPSIS	/usr/openwin/server/tsol/config.privs				
DESCRIPTION	<p>config.privs contains a list of all window privileges. config.privs lists each privilege in plain text, one per line, separated from the next by a new line. Lines preceded by a comment sign (#) are ignored.</p> <p>Each privilege not preceded by a comment overrides system checks for that privilege. The security administrator can comment out privileges in the list, but cannot add new privileges.</p> <p>By default, config.privs contains all the privileges that are allowed in the file: win_colormap, win_config, win_dga, win_devices, win_fontpath.</p> <p>config.privs should have a sensitivity label of ADMIN_LOW with permission bits 664, owner root, and group bin.</p>				
ATTRIBUTES	<p>See attributes(5) for descriptions of the following attributes:</p> <table> <tr> <th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr> <tr> <td>Availability</td><td>SUNWxwplt</td></tr> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWxwplt
ATTRIBUTE TYPE	ATTRIBUTE VALUE				
Availability	SUNWxwplt				
FILES	<p>/usr/openwin/server/tsol/config.privs</p> <p>List of window privileges that override system checks in the Trusted Solaris environment.</p>				
SEE ALSO					
Trusted Solaris 7 Reference Manual	priv_desc(4)				
SunOS 5.7 Reference Manual	attributes(5)				

NAME	device_allocate – Device allocate in permissions												
SYNOPSIS	/etc/security/device_allocate												
DESCRIPTION	<p>The <code>device_allocate</code> file contains mandatory access control information about each physical device. Each device is represented by a one-line entry of the form:</p> <p><i>device-name;device-type;device-minimum;device-maximum;device-authorization;device-clean</i></p> <p>where</p> <table> <tr> <td><i>device-name</i></td><td>This is an arbitrary text string naming the physical device. This field contains no embedded white space or non-printable characters.</td></tr> <tr> <td><i>device-type</i></td><td>This is an arbitrary text string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.</td></tr> <tr> <td><i>device-minimum</i></td><td>This is the minimum sensitivity label allowed for the device special files associated with the physical device. This field is a hex label.</td></tr> <tr> <td><i>device-maximum</i></td><td>This is the maximum sensitivity label allowed for the device special files associated with the physical device. This field is a hex label.</td></tr> <tr> <td><i>device-authorization</i></td><td>This is a comma-separated list of authorization numbers required to allocate the device, or an * to indicate that the device is not allocatable, or an @ to indicate that no explicit authorization is needed to allocate the device.</td></tr> <tr> <td><i>device-clean</i></td><td>This is the physical device's data purge program to be run any time the device is acted on by <code>allocate(1M)</code>. This is to ensure that all usable data is purged from the physical device before it is reused. This field contains the filename of a program in <code>/etc/security/lib</code>.</td></tr> </table> <p>The <code>device_allocate</code> file is a text file that resides in the <code>/etc/security</code> directory. The <code>device_allocate</code> file should not be edited by hand. The designated administrative role uses the Add Allocatable Device action to add a device and the Device Allocation Manager Configure dialog for modifications to</p>	<i>device-name</i>	This is an arbitrary text string naming the physical device. This field contains no embedded white space or non-printable characters.	<i>device-type</i>	This is an arbitrary text string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.	<i>device-minimum</i>	This is the minimum sensitivity label allowed for the device special files associated with the physical device. This field is a hex label.	<i>device-maximum</i>	This is the maximum sensitivity label allowed for the device special files associated with the physical device. This field is a hex label.	<i>device-authorization</i>	This is a comma-separated list of authorization numbers required to allocate the device, or an * to indicate that the device is not allocatable, or an @ to indicate that no explicit authorization is needed to allocate the device.	<i>device-clean</i>	This is the physical device's data purge program to be run any time the device is acted on by <code>allocate(1M)</code> . This is to ensure that all usable data is purged from the physical device before it is reused. This field contains the filename of a program in <code>/etc/security/lib</code> .
<i>device-name</i>	This is an arbitrary text string naming the physical device. This field contains no embedded white space or non-printable characters.												
<i>device-type</i>	This is an arbitrary text string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.												
<i>device-minimum</i>	This is the minimum sensitivity label allowed for the device special files associated with the physical device. This field is a hex label.												
<i>device-maximum</i>	This is the maximum sensitivity label allowed for the device special files associated with the physical device. This field is a hex label.												
<i>device-authorization</i>	This is a comma-separated list of authorization numbers required to allocate the device, or an * to indicate that the device is not allocatable, or an @ to indicate that no explicit authorization is needed to allocate the device.												
<i>device-clean</i>	This is the physical device's data purge program to be run any time the device is acted on by <code>allocate(1M)</code> . This is to ensure that all usable data is purged from the physical device before it is reused. This field contains the filename of a program in <code>/etc/security/lib</code> .												

a device. These tools preserve the desired file permissions, owner, group, and label, and audit all changes.

Lines in `device_allocate` can end with a `\\` to continue an entry on the next line.

Comments may also be included. A `#` makes a comment of all further text until the next `NEWLINE` not immediately preceded by a `\`.

Leading and trailing blanks are allowed in any of the fields.

The `device_allocate` file must be created by the system administrator before device allocation is enabled.

The `device_allocate` file is owned by `root`, with a group of `root`, a mode of `0644`, and a label of `ADMIN_LOW`.

EXAMPLES

EXAMPLE 1 Sample Device Allocate File

```
mag_tape_0; \
st;0x0000000000000000000000000000000000000000000000000000000000000000; \
0x7fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff; \
10;/etc/security/lib/st_clean floppy_0;fd; \
0x0000000000000000000000000000000000000000000000000000000000000000; \
0x7fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff; \
10;/etc/security/lib/disk_clean
```

FILES

`/etc/security/device_allocate` List of allocatable devices.

SUMMARY OF TRUSTED SOLARIS CHANGES

Devices are labeled, and by default require authorization for allocating and deallocating.

SEE ALSO

Trusted Solaris 7
Reference Manual

`allocate(1M)`, `deallocate(1M)`, `list_devices(1M)`

NAME	device_deallocate – Device deallocate file																																			
SYNOPSIS	/etc/security/tsol/device_deallocate																																			
DESCRIPTION	<p>The <code>device_deallocate</code> file contains device deallocation information for each physical device. Each device is represented by an entry of the form:</p> <p><i>device-name</i> ; <i>system-boot</i> ; <i>user-logout</i> ; <i>forced-logout</i> ;</p> <p>A backslash (\) at the end of a line continues the next line as part of the current entry. A pound sign (#) as the first character of a line indicates a comment line, which is ignored. Leading and trailing blanks are allowed in any of the fields.</p> <table><tr><td><i>device-name</i></td><td colspan="2">The name of the device.</td></tr><tr><td><i>system-boot</i></td><td colspan="2">Specifies what to do when the named device is found during system boot in an allocated state. This field may be one of these keywords:</td></tr><tr><td></td><td>FORCED_DEALLOCATE</td><td>Deallocate the device.</td></tr><tr><td></td><td>NO_ACTION</td><td>Leave the device in the allocated state.</td></tr><tr><td><i>user-logout</i></td><td colspan="2">Specifies what to do when the named device is allocated to a user who is logging out. This field may be one of these keywords:</td></tr><tr><td></td><td>ASK_USER</td><td>Ask the user whether to deallocate the device.</td></tr><tr><td></td><td>FORCED_DEALLOCATE</td><td>Deallocate the device.</td></tr><tr><td></td><td>NO_ACTION</td><td>Leave the device in the allocated state.</td></tr><tr><td><i>forced-logout</i></td><td colspan="2">Specifies what to do when the named device is allocated to a user who is being forced to log out. This field may be one of these keywords:</td></tr><tr><td></td><td>FORCED_DEALLOCATE</td><td>Deallocate the device.</td></tr><tr><td></td><td>NO_ACTION</td><td>Leave the device in the allocated state.</td></tr></table> <p><code>device_allocate</code> should be at a sensitivity label of <code>ADMIN_LOW</code> with permission bits 644, owner <code>root</code>, and group <code>sys</code>.</p>			<i>device-name</i>	The name of the device.		<i>system-boot</i>	Specifies what to do when the named device is found during system boot in an allocated state. This field may be one of these keywords:			FORCED_DEALLOCATE	Deallocate the device.		NO_ACTION	Leave the device in the allocated state.	<i>user-logout</i>	Specifies what to do when the named device is allocated to a user who is logging out. This field may be one of these keywords:			ASK_USER	Ask the user whether to deallocate the device.		FORCED_DEALLOCATE	Deallocate the device.		NO_ACTION	Leave the device in the allocated state.	<i>forced-logout</i>	Specifies what to do when the named device is allocated to a user who is being forced to log out. This field may be one of these keywords:			FORCED_DEALLOCATE	Deallocate the device.		NO_ACTION	Leave the device in the allocated state.
<i>device-name</i>	The name of the device.																																			
<i>system-boot</i>	Specifies what to do when the named device is found during system boot in an allocated state. This field may be one of these keywords:																																			
	FORCED_DEALLOCATE	Deallocate the device.																																		
	NO_ACTION	Leave the device in the allocated state.																																		
<i>user-logout</i>	Specifies what to do when the named device is allocated to a user who is logging out. This field may be one of these keywords:																																			
	ASK_USER	Ask the user whether to deallocate the device.																																		
	FORCED_DEALLOCATE	Deallocate the device.																																		
	NO_ACTION	Leave the device in the allocated state.																																		
<i>forced-logout</i>	Specifies what to do when the named device is allocated to a user who is being forced to log out. This field may be one of these keywords:																																			
	FORCED_DEALLOCATE	Deallocate the device.																																		
	NO_ACTION	Leave the device in the allocated state.																																		

Users may specify their own device deallocation options locally in their home directories in a file named `.device_deallocate`. The local file has the same format, and its entries take precedence; however, a local file's *system-boot* field is ignored and has no effect.

EXAMPLES

```
st0;\
FORCED_DEALLOCATE;\
NO_ACTION;\
NO_ACTION;\
# scsi tape
```

NOTES

The `device_deallocate` file is not read by `deallocate(1M)`. Rather, it is read by a GUI or a script program at the events of system boot, user logout, and forced logout. The program in term invokes the `deallocate` command when actual device deallocation is required.

FILES

```
/etc/security/tsol/device_deallocate
    Device deallocate file

$HOME/.device_deallocate
    User-customized device deallocate file
```

SEE ALSO

**Trusted Solaris 7
Reference Manual**

```
add_allocatable(1M), allocate(1M), deallocate(1M),
list_devices(1M), remove_allocatable(1M)
```

**SunOS 5.7 Reference
Manual**

```
attributes(5)
```

NAME	device_maps – Maps device names to physical devices
SYNOPSIS	<code>/etc/security/device_maps</code>
DESCRIPTION	<p>The <code>device_maps</code> file maps each physical device to a name and device type. Each device is represented by a one-line entry of the form:</p> <pre><i>device-name</i> : <i>device-type</i> : <i>device-list</i> :</pre> <p>where</p> <p><i>device-name</i> This is an arbitrary text string naming the physical device. This field contains no embedded white space or non-printable characters.</p> <p><i>device-type</i> This is an arbitrary text string naming the generic device type. This field identifies and groups together devices of like type. This field contains no embedded white space or non-printable characters.</p> <p><i>device-list</i> This is a list of the device special files associated with the physical device. This field contains valid device special file path names separated by white space.</p> <p>The <code>device_maps</code> file is a text file that resides in the <code>/etc/security/</code> directory.</p> <p>Lines in <code>device_maps</code> can end with a <code>\</code> to continue an entry on the next line.</p> <p>Comments may also be included. A <code>#</code> makes a comment of all further text until the next NEWLINE not immediately preceded by a <code>\</code>.</p> <p>Leading and trailing blanks are allowed in any of the fields.</p> <p>The <code>device_maps</code> file must be created by the system administrator before device allocation is enabled.</p> <p>This file is owned by <code>root</code>, with a group of <code>sys</code>, and a mode of <code>0644</code>.</p> <p><code>device_maps</code> have a label of <code>ADMIN_LOW</code> with permission bits <code>644</code>, owner <code>root</code>, and group <code>sys</code>.</p>
EXAMPLES	<p>EXAMPLE 1 A sample <code>device_maps</code> file</p> <pre># scsi tape st1:\ rmt:\</pre>


```
/dev/rst21 /dev/nrst21 /dev/rst5 /dev/nrst5 /dev/rst13 \  
/dev/nrst13 /dev/rst29 /dev/nrst29 /dev/rmt/1l /dev/rmt/1m \  
/dev/rmt/1 /dev/rmt/1h /dev/rmt/1u /dev/rmt/1ln /dev/rmt/1mn \  
/dev/rmt/1n /dev/rmt/1hn /dev/rmt/1un /dev/rmt/1b /dev/rmt/1bn:\
```

**SUMMARY
OF TRUSTED
SOLARIS
CHANGES**

The Trusted Solaris `device_maps` file is labeled, and is required to enable device allocation.

SEE ALSO
Trusted Solaris 7
Reference Manual

`allocate(1M)`, `deallocate(1M)`, `dminfo(1M)`, `list_devices(1M)`

NAME	device_policy – Device policy file												
DESCRIPTION	<p>The security policy for device files can differ from that for regular files and is configured through the <code>device_policy</code> database file. Rebooting the system in multiuser mode is required to effect the file's contents. Each entry in the file consists of one or more lines and represents the device policy configuration for one or more device files. A backslash (\) at the end of a line continues the next line as part of the current entry. A pound sign (#) as the first character of a line indicates a comment line, which is ignored. Each entry is of the form:</p> <pre>name:minor_name policy_type=value policy_type=value ...</pre> <p><i>name</i> is the name of a device driver.</p> <p><i>minor_name</i> is the actual name of a minor node, or a string of shell metacharacters that represent several minor nodes. See <code>sh(1)</code>.</p> <p>If two or more entries match a device, <code>devpolicy(1M)</code> uses the first matching entry. For example, for the following <code>device_policy</code> entries, the policy for <code>/dev/ptyp0</code> will differ from the policy for other <code>pty</code> devices.</p> <pre># # device_policy file # ptc: typ0 data_mac_policy=DR_MAC_EQ,DW_MAC_EQ # ptc:* data_mac_policy=DR_MAC_ANY,DW_MAC_ANY</pre> <p><i>policy_type=value</i> specifies a policy for the device nodes. There are four policy types: <code>data_mac_policy</code>, <code>attr_mac_policy</code>, <code>open_priv</code>, and <code>str_type</code>. The policy types and their allowed values are described below.</p> <p>data_mac_policy type This policy type specifies what a process's sensitivity label must be to have access to the device. The specified policy is enforced by the <code>open(2)</code> and <code>access(2)</code> system calls. The value for this type is a comma-separated pair of values: a read-MAC value and a write-MAC value:</p> <p>The read-MAC values are:</p> <table> <tbody> <tr> <td><code>DR_MAC_ANY</code></td><td>Process may have any SL.</td></tr> <tr> <td><code>DR_MAC_EQ</code></td><td>Process SL must be equal to device SL.</td></tr> <tr> <td><code>DR_MAC_NEQ</code></td><td>Process SL must not equal device SL.</td></tr> <tr> <td><code>DR_MAC_NEVER</code></td><td>Device is not read accessible.</td></tr> <tr> <td><code>DR_MAC_SDOM</code></td><td>Process SL must dominate device SL.</td></tr> <tr> <td><code>DR_MAC_ODOM</code></td><td>Process SL must be dominated by device SL.</td></tr> </tbody> </table> <p>The write-MAC values are:</p>	<code>DR_MAC_ANY</code>	Process may have any SL.	<code>DR_MAC_EQ</code>	Process SL must be equal to device SL.	<code>DR_MAC_NEQ</code>	Process SL must not equal device SL.	<code>DR_MAC_NEVER</code>	Device is not read accessible.	<code>DR_MAC_SDOM</code>	Process SL must dominate device SL.	<code>DR_MAC_ODOM</code>	Process SL must be dominated by device SL.
<code>DR_MAC_ANY</code>	Process may have any SL.												
<code>DR_MAC_EQ</code>	Process SL must be equal to device SL.												
<code>DR_MAC_NEQ</code>	Process SL must not equal device SL.												
<code>DR_MAC_NEVER</code>	Device is not read accessible.												
<code>DR_MAC_SDOM</code>	Process SL must dominate device SL.												
<code>DR_MAC_ODOM</code>	Process SL must be dominated by device SL.												

attr_mac_policy type

DW_MAC_ANY	Process may have any SL.
DW_MAC_EQ	Process SL must equal device SL.
DW_MAC_NEQ	Process SL must not equal device SL.
DW_MAC_NEVER	Device is not write accessible.
DW_MAC_SDOM	Process SL must dominate device SL.
DW_MAC_ODOM	Process SL must be dominated by device SL.

The optional read-MAC-modifier or write-MAC-modifier value is:
 MOD_AUTO_ALLOC Automatically allocate the device on behalf of the opening process.

The default policy is

data_mac_policy=DR_MAC_EQ,DW_MAC_EQ

This policy type specifies how to handle access to the device's attributes by the operations `acl(2)`, `chmod(2)`, `chown(2)`, and `stat(2)`. The value for this type is a comma-separated set of values: a read-MAC value, a write-MAC value, and an optional read-MAC modifier:

The read-MAC values are:

DR_MAC_ANY	Process may have any SL.
DR_MAC_EQ	Process SL must equal device SL.
DR_MAC_NEQ	Process SL must not equal device SL.
DR_MAC_NEVER	Device is not read accessible.
DR_MAC_SDOM	Process SL must dominate device SL.
DR_MAC_ODOM	Process SL must be dominated by device SL.

The write-MAC values are:

DW_MAC_ANY	Process may have any SL.
DW_MAC_EQ	Process SL must equal device SL.
DW_MAC_NEQ	Process SL must not equal device SL.
DW_MAC_NEVER	Device is not write accessible.
DW_MAC_SDOM	Process SL must dominate device SL.
DW_MAC_ODOM	Process SL must be dominated by device SL.

The optional read-MAC-modifier value is:

MOD_FABRICATE	Return fabricated device attributes to the reading process. Fabrication is designed for a process that
---------------	--

walks down an array of BSD-style pty's until it encounters an accessible pty (indicated by getting device attributes) or the end of the array.

The default policy is:

attr_mac_policy=DR_MAC_SDON,DW_MAC_EQ

open_priv type

This policy type specifies a privilege required to open the device. The specified privilege is required in addition to the data MAC policy. Privilege names can be in upper or lower case; or an integer ordinal can be used. For example,

open_priv=sys_devices

The default policy is:

open_priv=none

str_type type

The streams type, meaningful only for streams devices, specifies how the kernel streams head should control streams messages. The value can be one of these keywords:

DSTR_LOOP

Loop type stream. Unlabeled streams control messages are allowed. Unlabeled data messages are not allowed.

DSTR_NET

Network type Stream. Unlabeled Stream messages are not allowed.

DSTR_DEV

Device type Stream. Unlabeled Stream messages are allowed.

An example is:

str_type=DSTR_NET

The default policy is:

str_type=STR_DEV

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsu

EXAMPLES**EXAMPLE 1** A complete policy — Sample

```
mm:kmem \  
data_mac_policy=DR_MAC_EQ,DW_MAC_EQ \  
attr_mac_policy=DR_MAC_SDOM,DW_MAC_EQ  
  
mm:null \  
data_mac_policy=DR_MAC_ANY,DW_MAC_ANY \  
attr_mac_policy=DR_MAC_SDOM,DW_MAC_EQ
```

FILES

/etc/security/tsol/device_policy Device policy file.

SEE ALSO

Trusted Solaris 7
Reference Manual

devpolicy(1M)

SunOS 5.7 Reference
Manual

sh(1), attributes(5)

NAME	inetd.conf – Internet servers database										
SYNOPSIS	<pre>/etc/inet/inetd.conf /etc/inetd.conf</pre>										
DESCRIPTION	<p>The <code>inetd.conf</code> file contains the list of servers that <code>inetd(1M)</code> invokes when it receives an Internet request over a socket. Each server entry is composed of a single line of the form:</p> <pre><i>service-name endpoint-type protocol wait-status uid server-program \ server-arguments</i></pre> <p>Fields are separated by either SPACE or TAB characters. A # (number sign) indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines that search this file.</p> <p><i>service-name</i> The name of a valid service listed in the <code>services</code> file. For RPC services, the value of the <i>service-name</i> field consists of the RPC service name or program number, followed by a / (slash) and either a version number or a range of version numbers (for example, <code>rstatd/2-4</code>).</p> <p><i>endpoint-type</i> Can be one of:</p> <table> <tr> <td><code>stream</code></td><td>For a stream socket</td></tr> <tr> <td><code>dgram</code></td><td>For a datagram socket</td></tr> <tr> <td><code>raw</code></td><td>For a raw socket</td></tr> <tr> <td><code>seqpacket</code></td><td>For a sequenced packet socket</td></tr> <tr> <td><code>tli</code></td><td>For all TLI endpoints</td></tr> </table> <p><i>protocol</i> Must be a recognized protocol listed in the file <code>/etc/inet/protocols</code>. For RPC services, the field consists of the string <code>rpc</code> followed by a / (slash) and either a * (asterisk), one or more nettypes, one or more netids, or a combination of nettypes and netids. Whatever the value, it is first treated as a nettype. If it is not a valid nettype, then it is treated as a netid. For example, <code>rpc/*</code> for an RPC service using all the transports supported by the system (the list can be found in the <code>/etc/netconfig</code> file), equivalent to saying</p>	<code>stream</code>	For a stream socket	<code>dgram</code>	For a datagram socket	<code>raw</code>	For a raw socket	<code>seqpacket</code>	For a sequenced packet socket	<code>tli</code>	For all TLI endpoints
<code>stream</code>	For a stream socket										
<code>dgram</code>	For a datagram socket										
<code>raw</code>	For a raw socket										
<code>seqpacket</code>	For a sequenced packet socket										
<code>tli</code>	For all TLI endpoints										

wait-status

`rpc/visible rpc/ticots` for an RPC service using the Connection-Oriented Transport Service.

`nowait` for all but “single-threaded” datagram servers — servers which do not release the socket until a timeout occurs. These must have the `status wait`. Do not configure `udp` services as `nowait`. This will cause a race condition where the `inetd` program selects on the socket and the server program reads from the socket. Many server programs will be forked and performance will be severely compromised.

A new option exists for `udp` servers. The `-poly` option, is similar to the `-wait` option except that `-poly` allows a separate server to be started at each sensitivity label. This option is allowed only for `udp` servers.

If the server program should inherit the trusted path attribute, the *wait-status* field should include the keyword `trusted`, separated from other keywords in the field by a comma. If the keyword is not present, the trusted path attribute will not be propagated to the server.

If the server program should inherit audit characteristics from the client, the *wait-status* field should include the keyword `setaudit`, separated from other keywords in the field by a comma. If the `setaudit` keyword is present, the audit ID, audit terminal ID, and audit preselection mask of the client will be transferred to the server.

uid

The user ID under which the server should run. This allows servers to run with access privileges other than those for root. If the server should run with the ID of the client making the call to the server, a keyword of `CLIENT` should be entered in the *uid* field. The `CLIENT` keyword is allowed only for `nowait` servers. If the `CLIENT` keyword is present the user ID, group ID, and supplementary groups of the client will be transferred to the server.

SUMMARY OF TRUSTED SOLARIS CHANGES

server-program Either the pathname of a server program to be invoked by *inetd* to perform the requested service, or the value *internal* if *inetd* itself provides the service.

server-arguments If a server must be invoked with command line arguments, the entire command line (including argument 0) must appear in this field (which consists of all remaining words in the entry). If the server expects *inetd* to pass it the address of its peer (for compatibility with 4.2BSD executable daemons), then the first argument to the command should be specified as '%A'. No more than five arguments are allowed in this field.

The *wait-status* field is extended to allow a *trusted* keyword to specify that the trusted path attribute should be passed to the server by *inetd*. If you want a server to run with the audit characteristics of the client, the *wait-status* field can now contain a keyword of *setaudit*.

If you want a *nowait* server to run with the user ID of the client, the *uid* field can now contain a keyword of *CLIENT*.

The *-poly* option has been added for *udp* servers. The option is similar to the *-wait* option except that *-poly* allows a separate server to be started at each sensitivity label.

FILES

<i>/etc/netconfig</i>	Network configuration file
<i>/etc/inet/protocols</i>	Internet protocols
<i>/etc/inet/services</i>	Internet network services

SEE ALSO

Trusted Solaris 7
Reference Manual

in.tftpd(1M), *inetd(1M)*

SunOS 5.7 Reference
Manual

rlogin(1), *rsh(1)*, *services(4)*

NOTES

/etc/inet/inetd.conf is the official SVR4 name of the *inetd.conf* file. The symbolic link */etc/inetd.conf* exists for BSD compatibility.

NAME	inittab – Script for init
DESCRIPTION	<p>The file <code>/etc/inittab</code> controls process dispatching by <code>init</code>. The processes most typically dispatched by <code>init</code> are daemons.</p> <p>The <code>inittab</code> file is composed of entries that are position dependent and have the following format:</p> <pre><i>id</i>:<i>rstate</i>:<i>action</i>:<i>process</i></pre> <p>Each entry is delimited by a newline; however, a backslash (<code>\</code>) preceding a newline indicates a continuation of the entry. Up to 512 characters for each entry are permitted. Comments may be inserted in the <i>process</i> field using the convention for comments described in <code>sysh(1M)</code>. . There are no limits (other than maximum entry size) imposed on the number of entries in the <code>inittab</code> file. The entry fields are:</p> <p><i>id</i></p> <p>One or two characters used to uniquely identify an entry.</p> <p><i>rstate</i></p> <p>Define the run level in which this entry is to be processed. Run-levels effectively correspond to a configuration of processes in the system. That is, each process spawned by <code>init</code> is assigned a run level(s) in which it is allowed to exist. The run levels are represented by a number ranging from 0 through 6. For example, if the system is in run level 1, only those entries having a 1 in the <i>rstate</i> field are processed.</p> <p>When <code>init</code> is requested to change run levels, all processes that do not have an entry in the <i>rstate</i> field for the target run level are sent the warning signal <code>SIGTERM</code> and allowed a 5-second grace period before being forcibly terminated by the kill signal <code>SIGKILL</code>. The <i>rstate</i> field can define multiple run levels for a process by selecting more than one run level in any combination from 0 through 6. If no run level is specified, then the process is assumed to be valid at all run levels 0 through 6.</p> <p>There are three other values, <code>a</code>, <code>b</code> and <code>c</code>, which can appear in the <i>rstate</i> field, even though they are not true run levels. Entries which have these characters in the <i>rstate</i> field are processed only when an <code>init</code> or <code>telinit</code> process requests them to be run (regardless of the current run level of the system). See <code>init(1M)</code>. These differ from run levels in that <code>init</code> can never enter run level <code>a</code>, <code>b</code> or <code>c</code>. Also, a request for the execution of any of these processes does not change the current run level. Furthermore, a process started by an <code>a</code>, <code>b</code> or <code>c</code> command is not killed when <code>init</code> changes levels. They are killed only if their line in <code>inittab</code> is marked <code>off</code> in the <i>action</i> field, their line is deleted entirely from <code>inittab</code>, or <code>init</code> goes into single-user state.</p>

action

Key words in this field tell `init` how to treat the process specified in the *process* field. The actions recognized by `init` are as follows:

respawn

If the process does not exist, then start the process; do not wait for its termination (continue scanning the `inittab` file), and when the process dies, restart the process. If the process currently exists, do nothing and continue scanning the `inittab` file.

wait

When `init` enters the run level that matches the entry's *rstate*, start the process and wait for its termination. All subsequent reads of the `inittab` file while `init` is in the same run level cause `init` to ignore this entry.

once

When `init` enters a run level that matches the entry's *rstate*, start the process, do not wait for its termination. When it dies, do not restart the process. If `init` enters a new run level and the process is still running from a previous run level change, the program is not restarted.

boot

The entry is to be processed only at `init`'s boot-time read of the `inittab` file. `init` is to start the process and not wait for its termination; when it dies, it does not restart the process. In order for this instruction to be meaningful, the *rstate* should be the default or it must match `init`'s run level at boot time. This action is useful for an initialization function following a hardware reboot of the system.

bootwait

The entry is to be processed the first time `init` goes from single-user to multi-user state after the system is booted. (If `initdefault` is set to 2, the process runs right after the boot.) `init` starts the process, waits for its termination and, when it dies, does not restart the process.

powerfail

Execute the process associated with this entry only when `init` receives a power fail signal, `SIGPWR` (see `signal(3C)`).

powerwait

Execute the process associated with this entry only when `init` receives a power fail signal, `SIGPWR`, and wait until it terminates before continuing any processing of `inittab`.

off

If the process associated with this entry is currently running, send the warning signal `SIGTERM` and wait 5 seconds before forcibly terminating the process with the kill signal `SIGKILL`. If the process is nonexistent, ignore the entry.

ondemand

This instruction is really a synonym for the `respawn` action. It is functionally identical to `respawn` but is given a different keyword in order to divorce its association with run levels. This instruction is used only with the `a`, `b` or `c` values described in the `rstate` field.

initdefault

An entry with this action is scanned only when `init` is initially invoked. `init` uses this entry to determine which run level to enter initially. It does this by taking the highest run level specified in the `rstate` field and using that as its initial state. If the `rstate` field is empty, this is interpreted as `0123456` and `init` will enter run level 6. This will cause the system to loop (it will go to firmware and reboot continuously). Additionally, if `init` does not find an `initdefault` entry in `inittab`, it requests an initial run level from the user at reboot time.

sysinit

Entries of this type are executed before `init` tries to access the console (that is, before the Console Login: prompt). It is expected that this entry will be used only to initialize devices that `init` might try to ask the run level question. These entries are executed and `init` waits for their completion before continuing.

process

Specify a command to be executed. The entire `process` field is prefixed with `exec` and passed to a forked `sh` as `sh -c 'exec command'`. For this reason, any legal `sh` syntax can appear in the `process` field.

The Trusted Solaris environment uses the `sysh` shell.

**SUMMARY
OF TRUSTED
SOLARIS
CHANGES**

SEE ALSO

Trusted Solaris 7
Reference Manual

SunOS 5.7 Reference
Manual

`init(1M)`, `sysh(1M)`, `exec(2)`, `open(2)`

`who(1)`, `ttymon(1M)`, `signal(3C)`

NAME	label_encodings – Label encodings file
SYNOPSIS	/etc/security/tsol/label_encodings
DESCRIPTION	<p>In addition to the required sections of the label encodings file described in <i>Compartmented Mode Workstation Labeling: Encodings Format</i>, the Trusted Solaris environment accepts optional local extensions. These extensions provide various translation options and an association between character-coded color names and sensitivity labels.</p> <p>The optional local extensions section starts with the <code>LOCAL DEFINITIONS:</code> keyword and is followed by zero or more of the following unordered statements:</p> <p><code>ADMIN LOW NAME=<i>name</i></code></p> <p>The string <i>name</i> is accepted as an alternate name for the <code>ADMIN_LOW</code> label when translating from character-coded to binary form. The string <i>name</i> is the string returned when translating the <code>ADMIN_LOW</code> label from binary to character-coded form. If this option is not specified, <code>ADMIN_LOW</code> is used.</p> <p>Note that use of this option could lead to interoperability problems with machines which do not have the same alternate name.</p> <p><code>ADMIN HIGH NAME=<i>name</i></code></p> <p>The string <i>name</i> is accepted as an alternate name for the <code>ADMIN_HIGH</code> label when translating from character-coded form to binary form. The string <i>name</i> is the string returned when translating the <code>ADMIN_HIGH</code> label from binary to character-coded form. If this option is not specified, <code>ADMIN_HIGH</code> is used.</p> <p>Note that use of this option could lead to interoperability problems with machines which do not have the same alternate name.</p> <p><code>DEFAULT LABEL VIEW IS EXTERNAL</code></p> <p>Unless otherwise specified, when an <code>ADMIN_HIGH</code> or <code>ADMIN_LOW</code> binary label is translated to a character-coded label, the character-coded label will be in external form. In external form <code>ADMIN_HIGH</code> is demoted to the maximum label and <code>ADMIN_LOW</code> is promoted to the minimum label. External label view is the default condition.</p> <p><code>DEFAULT LABEL VIEW IS INTERNAL</code></p> <p>Unless otherwise specified, when an <code>ADMIN_HIGH</code> or <code>ADMIN_LOW</code> binary label is translated to a character-coded label, the character-coded label will be in internal form. In internal form, <code>ADMIN_HIGH</code> is represented by the string <code>ADMIN_HIGH</code> and <code>ADMIN_LOW</code> is represented by the string <code>ADMIN_LOW</code>.</p> <p><code>DEFAULT FLAGS= <i>value</i></code></p> <p>This option represents a default GFI <code>Flags=</code> keyword value to be used if no flags are specified as a parameter to the translation. Caution must</p>

be taken when defining a `DEFAULT FLAGS= value` that the appropriate `Flags= values` have been provided. A non-zero value also implies that label validation during translation from binary to character-coded form is not done. The default value is 0 (zero).

`FORCED FLAGS= value`

This option represents a `GFI Flags= keyword value` to be used in all translations. Caution must be taken when defining a `FORCED FLAGS= value` that the appropriate `Flags= values` have been provided. A non-zero value also implies that label validation during translation from binary to character-coded form is not done. The default value is 0 (zero).

`CLASSIFICATION NAME= name`

This option specifies the string to be displayed in the Label builder GUI for the title of the Classification names section. Specifying a `NULL` value for *name* leaves the section without a title. The default value is `CLASSIFICATION`.

`COMPARTMENTS NAME= name`

This option specifies the string to be displayed in the label builder GUI for the title of the *Compartment Word* section. Specifying a `NULL` value for *name* leaves the section without a title. The default value is `COMPARTMENTS`.

The final part of the `LOCAL DEFINITIONS:` section defines the character-coded color names to be associated with various words, sensitivity labels, or classifications. This section supports the `bltocolor(3)` function. It consists of the `COLOR NAMES:` keyword and is followed by zero or more color-to-label assignments. Each statement has one of the following two syntaxes:

`word= word value; color= color value;`

`label= label value; color= color value;`

where *color value* is a character-coded color name to be associated with the word *word value*, sensitivity label *label value*, or classification *label value*.

The character-coded color name *color value* for a label is determined by the order of entries in the `COLOR NAMES:` section that make up the label. If a label contains a word *word value* that is specified in this section, the *color value* of the label is the one associated with the first *word value* specified. If no specified word *word value* is contained in the label, the *color value* is the one associated with an exact match of a *label value*. If there is no exact match, the *color value* is the one associated with the first specified *label value* whose classification matches the classification of the label.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsr

EXAMPLES**EXAMPLE 1** A sample LOCAL DEFINITIONS: section

```

LOCAL DEFINITIONS:
ADMIN LOW NAME= LoLo;  * It is strongly advised not to use this option
ADMIN HIGH NAME= HiHi;  * It is strongly advised not to use this option

DEFAULT LABEL VIEW IS INTERNAL;

DEFAULT FLAGS= 0x4;
FORCED FLAGS= 0;

CLASSIFICATION NAME=;  * No Classification name title
COMPARTMENTS NAME=;  * No Compartments word title

COLOR NAMES:

label= Admin_Low; color= Pale Blue;
label= unclassified; color= light grey;
word= Project A; color= bright blue;
label= c; color= sea foam green;
label= secret; color= #ff0000; * Hexadecimal RGB value
word= Hotel; color= Lavender;
word= KeLO; color= red;
label= TS; color= khaki;
label= TS Elephant; color= yellow;
label= Admin_High; color= shocking pink;

```

FILES

/etc/security/tsol/label_encodings

The label encodings file contains the classification names, words, constraints, and values for the defined labels of this system.

DIAGNOSTICS

The following diagnostics are in addition to those found in Appendix A of *Compartmented Mode Workstation Labeling: Encodings Format*:

Admin_High color already assigned as “XXX”.

A color has already been defined for the ADMIN_HIGH label. Another cannot be defined.

Admin_Low color already assigned as “XXX”.

A color has already been defined for the ADMIN_LOW label. Another cannot be defined.

BOUND TRANSLATION BY CLEARANCE obsolete, Bound is always a Sensitivity Label.

This option is obsolete and ignored. All label translations are bound by the calling process' sensitivity label.

Can't allocate NNN bytes for ADMIN HIGH NAME=

The system cannot dynamically allocate the memory it needs to process the ADMIN_HIGH NAME= option.

Can't allocate NNN bytes for ADMIN LOW NAME=

The system cannot dynamically allocate the memory it needs to process the ADMIN_LOW NAME= option.

Can't allocate NNN bytes for CLASSIFICATION NAME=

The system cannot dynamically allocate the memory it needs to process the CLASSIFICATION NAME= option.

Can't allocate NNN bytes for COMPARTMENTS NAME=

The system cannot dynamically allocate the memory it needs to process the COMPARTMENTS NAME= option.

Can't allocate NNN bytes for color name "XXX".

The system cannot dynamically allocate the memory it needs to store color name XXX.

Can't allocate NNN bytes for color names table.

The system cannot dynamically allocate the memory it needs to process the COLOR NAMES: section.

Can't allocate NNN bytes for color table entry.

The system cannot dynamically allocate the memory it needs to process a Color Table entry.

Can't allocate NNN bytes for color word entry.

The system cannot dynamically allocate the memory it needs to process a Color Word entry.

Duplicate ADMIN HIGH NAME= ignored.

More than one ADMIN HIGH NAME= option was encountered. All but the first are ignored.

Duplicate ADMIN LOW NAME= ignored.

More than one ADMIN LOW NAME= option was encountered. All but the first are ignored.

Duplicate CLASSIFICATION NAME= ignored.

More than one CLASSIFICATION NAME= option was encountered.
All but the first are ignored.

Duplicate COMPARTMENTS NAME= ignored.

More than one COMPARTMENTS NAME= option was encountered.
All but the first are ignored.

End of File or LOCAL DEFINITIONS: not found. Found instead: "XXX".

The noted extraneous text was found when the LOCAL
DEFINITIONS: section or end of label encodings file was
expected.

Found color "XXX" without associated label.

The color XXX was found, however it had no label or word
associated with it.

Invalid color label "XXX".

The label XXX cannot be parsed.

Label preceding "XXX" did not have a color specification.

A label or word was found without a matching color name.

Word "XXX" not found as a valid Sensitivity Label word.

The word XXX was not found as a valid word for a
sensitivity label.

SEE ALSO

**Trusted Solaris 7
Reference Manual**

bcltobanner(3), blinset(3), bltocolo(3), bltos(3), blvalid(3),
labelinfo(3), labelvers(3), stobl(3), chk_encodings(1M)

Trusted Solaris Label Administration

Defense Intelligence Agency document DDS-2600-6216-93, *Compartmented Mode
Workstation Labeling: Encodings Format*, September 1993.

**SunOS 5.7 Reference
Manual**

attributes(5)

WARNINGS

Creation of and modification to the label encodings file should only be
undertaken with a thorough understanding not only of the concepts in
Compartmented Mode Workstation Labeling: Encodings Format but also of the
details of the local labeling requirements.

The following warnings are paraphrased from *Compartmented Mode Workstation
Labeling: Encodings Format*.

Take extreme care when modifying a label encodings file that is already loaded and running in a Trusted Solaris environment. Once the system runs with the label encodings file, many objects are labeled with sensitivity labels that are well formed with respect to the loaded label encodings file. If the label encodings file is subsequently changed, it is possible that the existing labels will no longer be well-formed. Changing the bit patterns associated with words causes existing objects whose labels contain the words to have possibly invalid labels. Raising the minimum classification or lowering the maximum classification associated with words will likely cause existing objects whose labels contain the words to no longer be well-formed.

Information Labels (ILs) are now obsolete. See NOTES.

Changes to a current encodings file that has already been used should be limited only to adding new classifications or words, changing the names of existing words, or modifying the local extensions. As described in *Compartmented Mode Workstation Labeling: Encodings Format*, it is important to reserve extra inverse bits when the label encodings file is first created to allow for later expansion of the label encodings file to incorporate new inverse words. If an inverse word is added that does not use reserved inverse bits, all existing objects in the environment will erroneously have labels that include the new inverse word.

NOTES

Defining the label encodings file is a three-step process. First, the set of human-readable labels to be represented must be identified and understood. The definition of this set includes the list of classifications and other words used in the human-readable labels, relations between and among the words, classification restrictions associated with use of each word, and intended use of the words in mandatory access control and labeling system output. Next, this definition is associated with an internal format of integers, bit patterns, and logical relationship statements. Finally, a label encodings file is created. The *Compartmented Mode Workstation Labeling: Encodings Format* document describes the second and third steps, and assumes that the first has already been performed.

Information labels (ILs) are not supported in Trusted Solaris 7 and later releases. Trusted Solaris software interprets any ILs on communications and files from systems running earlier releases as ADMIN_LOW.

Objects still have CMW labels, and CMW labels still include the IL component: IL[SL]; however, the IL component is fixed at ADMIN_LOW.

As a result, Trusted Solaris 7 has the following characteristics:

- ILs do not display in window labels; SLs (Sensitivity Labels) display alone within brackets.
- ILs do not float.
- Setting an IL on an object has no effect.

- Getting an object's IL will always return `ADMIN_LOW`.
- Although certain utilities, library functions, and system calls can manipulate IL strings, the resulting ILs are always `ADMIN_LOW`, and cannot be set on any objects.
- Options related to information labels in the `label_encodings(4)` file can be ignored:

```
Markings Name= Marks;  
Float Process Information Label;
```

NAME	mnttab – Mounted file system table	
DESCRIPTION	<p>The file <code>mnttab</code> resides in <code>/etc</code> and contains information about devices that are <i>currently</i> mounted. <code>mnttab</code> is read by programs using the routines described in <code>getmntent(3C)</code>. <code>mount(1M)</code> adds entries to this file. <code>umount</code> removes entries from this file. Each entry is a line of fields separated by spaces in the form:</p> <pre><i>special mount_point fstype options time</i></pre> <p>where</p> <p><i>special</i> The name of the resource to be mounted.</p> <p><i>mount_point</i> The pathname of the directory on which the filesystem is mounted.</p> <p><i>fstype</i> The file system type of the mounted file system.</p> <p><i>options</i> The mount options. (See respective mount filesystem man page below in SEE ALSO.)</p> <p><i>time</i> The time at which the file system was mounted.</p> <p>Examples of entries for the <i>special</i> field include the pathname of a block-special device, the name of a remote filesystem in <i>host:pathname</i> form, or the name of a “swap file” (for instance, a file made with <code>mkfile(1M)</code>).</p>	
SUMMARY OF TRUSTED SOLARIS CHANGES	<p>The <code>/etc/mnttab</code> file must have a sensitivity label of <code>ADMIN_LOW</code> and an owner UID of 0.</p>	
FILES	<code>/etc/mnttab</code>	Mounted file system table.
SEE ALSO		
Trusted Solaris 7 Reference Manual	<code>mount_hsf(1M)</code> , <code>mount_nfs(1M)</code> , <code>mount_pcfs(1M)</code> , <code>mount_ufs(1M)</code> , <code>mount(1M)</code> , <code>setmnt(1M)</code>	
SunOS 5.7 Reference Manual	<code>mkfile(1M)</code> , <code>getmntent(3C)</code>	

NAME	nsswitch.conf – Configuration file for the name service switch																																												
SYNOPSIS	/etc/nsswitch.conf																																												
DESCRIPTION	<p>The operating environment uses a number of "databases" of information about hosts, users (passwd/shadow), groups and so forth. Data for these can come from a variety of sources: host-names and host-addresses, for example, may be found in /etc/hosts, NIS+, or DNS. Zero or more sources may be used for each database; the sources and their lookup order are specified in the /etc/nsswitch.conf file.</p> <p>The following databases use the switch file:</p> <table> <tr> <th><i>Database</i></th><th><i>Used by</i></th></tr> <tr> <td>aliases</td><td>sendmail(1M)</td></tr> <tr> <td>automount</td><td>automount(1M)</td></tr> <tr> <td>bootparams</td><td>rpc.bootparamd(1M)</td></tr> <tr> <td>ethers</td><td>ethers(3N)</td></tr> <tr> <td>group</td><td>getgrnam(3C)</td></tr> <tr> <td>hosts</td><td>gethostbyname(3N)</td></tr> <tr> <td></td><td>(See "Interaction with netconfig" below.)</td></tr> <tr> <td>netgroup</td><td>innetgr(3N)</td></tr> <tr> <td>netmasks</td><td>ifconfig(1M)</td></tr> <tr> <td>networks</td><td>getnetbyname(3N)</td></tr> <tr> <td>passwd</td><td>getpwnam(3C), getspnam(3C)</td></tr> <tr> <td>protocols</td><td>getprotobyname(3N)</td></tr> <tr> <td>publickey</td><td>getpublickey(3N) secure_rpc(3N)</td></tr> <tr> <td>rpc</td><td>getrpcbyname(3N)</td></tr> <tr> <td>sendmailvars</td><td>sendmail(1M)</td></tr> <tr> <td>services</td><td>getservbyname(3N)</td></tr> <tr> <td></td><td>(See "Interaction with netconfig" below.)</td></tr> <tr> <td>tsolprof</td><td>tsolprof(4)</td></tr> <tr> <td>tsoluser</td><td>tsoluser(4)</td></tr> <tr> <td>tnrhdb</td><td>tnrhdb(4)</td></tr> <tr> <td>tnrhttp</td><td>tnrhttp(4)</td></tr> </table>	<i>Database</i>	<i>Used by</i>	aliases	sendmail(1M)	automount	automount(1M)	bootparams	rpc.bootparamd(1M)	ethers	ethers(3N)	group	getgrnam(3C)	hosts	gethostbyname(3N)		(See "Interaction with netconfig" below.)	netgroup	innetgr(3N)	netmasks	ifconfig(1M)	networks	getnetbyname(3N)	passwd	getpwnam(3C), getspnam(3C)	protocols	getprotobyname(3N)	publickey	getpublickey(3N) secure_rpc(3N)	rpc	getrpcbyname(3N)	sendmailvars	sendmail(1M)	services	getservbyname(3N)		(See "Interaction with netconfig" below.)	tsolprof	tsolprof(4)	tsoluser	tsoluser(4)	tnrhdb	tnrhdb(4)	tnrhttp	tnrhttp(4)
<i>Database</i>	<i>Used by</i>																																												
aliases	sendmail(1M)																																												
automount	automount(1M)																																												
bootparams	rpc.bootparamd(1M)																																												
ethers	ethers(3N)																																												
group	getgrnam(3C)																																												
hosts	gethostbyname(3N)																																												
	(See "Interaction with netconfig" below.)																																												
netgroup	innetgr(3N)																																												
netmasks	ifconfig(1M)																																												
networks	getnetbyname(3N)																																												
passwd	getpwnam(3C), getspnam(3C)																																												
protocols	getprotobyname(3N)																																												
publickey	getpublickey(3N) secure_rpc(3N)																																												
rpc	getrpcbyname(3N)																																												
sendmailvars	sendmail(1M)																																												
services	getservbyname(3N)																																												
	(See "Interaction with netconfig" below.)																																												
tsolprof	tsolprof(4)																																												
tsoluser	tsoluser(4)																																												
tnrhdb	tnrhdb(4)																																												
tnrhttp	tnrhttp(4)																																												

tsolprof	tsolprof(4)
tsoluser	tsoluser(4)
tnrhdb	tnrhdb(4)
tnrhttp	tnrhttp(4)

The following sources may be used:

<i>Source</i>	<i>Uses</i>
files	/etc/hosts, /etc/passwd, /etc/shadow and so forth
nisplus	NIS+
dns	Valid only for hosts; uses the Internet Domain Name Service.
compat	Valid only for passwd and group; implements + and -. (See "Interaction with +/- syntax" below.) The compat source may not be supported in future releases.

There is an entry in /etc/nsswitch.conf for each database. Typically these entries will be simple, such as "protocols: files" or "networks: files nisplus". However, when multiple sources are specified, it is sometimes necessary to define precisely the circumstances under which each source will be tried. A source can return one of the following codes:

<i>Status</i>	<i>Meaning</i>
SUCCESS	Requested database entry was found.
UNAVAIL	Source is not responding or is corrupted.
NOTFOUND	Source responded "no such entry".
TRYAGAIN	Source is busy, might respond to retries.

For each status code, two actions are possible:

<i>Action</i>	<i>Meaning</i>
continue	Try the next source in the list.
return	Return now.

The complete syntax of an entry is

```
<entry>      ::= <database> ":" [<source>
[<criteria>]]*
<criteria>   ::= "[" <criterion>+ "]"
```

```

<criterion> ::= <status> "=" <action>
<status>    ::= "success" | "notfound" | "unavail" | "tryagain"
<action>    ::= "return" | "continue"

```

Each entry occupies a single line in the file. Lines that are blank, or that start with white space, are ignored. Everything on a line following a # character is also ignored; the # character can begin anywhere in a line, to be used to begin comments. The <database> and <source> names are case-sensitive, but <action> and <status> names are case-insensitive.

The library functions contain compiled-in default entries that are used if the appropriate entry in `nsswitch.conf` is absent or syntactically incorrect.

The default criteria are to continue on anything except SUCCESS; in other words, [SUCCESS=return NOTFOUND=continue UNAVAIL=continue TRYAGAIN=continue].

The default, or explicitly specified, criteria are meaningless following the last source in an entry; and they are ignored, since the action is always to return to the caller irrespective of the status code the source returns.

Interaction with netconfig

In order to ensure that they all return consistent results, `gethostbyname(3N)`, `getservbyname(3N)`, and `netdir_getbyname(3N)` functions are all implemented in terms of the same internal library function. This function obtains the system-wide source lookup policy for `hosts` and `services` based on the `inet` family entries in `netconfig(4)` and uses the switch entries only if the `netconfig` entries have a "-" in the last column for `nametoaddr` libraries. See the NOTES section in `gethostbyname(3N)` and `getservbyname(3N)` for details.

Interaction with Password Aging

When password aging is turned on, only a limited set of possible name services are permitted for the `passwd:` *database* in the `/etc/nsswitch.conf` file:

```

passwd:                files

passwd:                files nisplus

passwd:                compat

passwd_compat:         nisplus

```

Any other settings will cause the `passwd(1)` command to fail when it attempts to change the password after expiration and will prevent the user from logging in. These are the *only* permitted settings when password aging has been turned on. Otherwise, you can work around incorrect `passwd:` lines by using the `-r repository` argument to the `passwd(1)` command and using `passwd -r repository` to override the `nsswitch.conf` settings and specify in which name service you want to modify your password.

**Interaction with +/-
syntax**

Releases prior to SunOS 5.0 did not have the name service switch but did allow the user some policy control. In `/etc/passwd` one could have entries of the form `+user` (include the specified user from NIS `passwd.byname`), `-user` (exclude the specified user) and `+` (include everything, except excluded users, from NIS `passwd.byname`). The desired behavior was often "everything in the file followed by everything in NIS", expressed by a solitary `+` at the end of `/etc/passwd`. The switch provides an alternative for this case ("`passwd: files nis`") that does not require `+` entries in `/etc/passwd` and `/etc/shadow` (the latter is a new addition to SunOS 5.0, see `shadow(4)`).

If this is not sufficient, the NIS/YP compatibility source provides full `+/-` semantics. It reads `/etc/passwd` for `getpwnam(3C)` functions and `/etc/shadow` for `getspnam(3C)` functions and, if it finds `+/-` entries, invokes an appropriate source. By default, the source is "nis", but this may be overridden by specifying "nisplus" as the source for the pseudo-database `passwd_compat`.

Note that for every `/etc/passwd` entry, there should be a corresponding entry in the `/etc/shadow` file.

**Useful
Configurations**

The compiled-in default entries for all databases use NIS (YP) as the enterprise level name service and are identical to those in the default configuration of this file:

```
passwd:          files nis
group:           files nis
hosts:           nis [NOTFOUND=return] files
networks:        nis [NOTFOUND=return] files
protocols:       nis [NOTFOUND=return] files
rpc:             nis [NOTFOUND=return] files
ethers:          nis [NOTFOUND=return] files
netmasks:        nis [NOTFOUND=return] files
bootparams:      nis [NOTFOUND=return] files
publickey:       nis [NOTFOUND=return] files
netgroup:        nis
automount:       files nis
aliases:         files nis
services:        files nis
sendmailvars:    files
```

The policy "nis [NOTFOUND=return] files" implies "if nis is UNAVAIL, continue on to files, and if nis returns NOTFOUND, return to the caller; in other words, treat nis as the authoritative source of information and try files only if nis is down. This, and other policies listed in the default configuration above, are identical to the hard-wired policies in SunOS releases prior to 5.0.

If compatibility with the +/- syntax for passwd and group is required, simply modify the entries for passwd and group to:

```
passwd:                compat
```

```
group:                 compat
```

If NIS+ is the enterprise level name service, the default configuration should be modified to use nisplus instead of nis for every database on client machines. The file /etc/nsswitch.nisplus contains a sample configuration that can be copied to /etc/nsswitch.conf to set this policy.

If the use of +/- syntax is desired in conjunction with nisplus, use the following four entries:

```
passwd:                compat
```

```
passwd_compat:         nisplus
```

```
group:                 compat
```

```
group_compat:          nisplus
```

In order to get information from the Internet Domain Name Service for hosts that are not listed in the enterprise level name service, NIS+, use the following configuration and set up the /etc/resolv.conf file (see resolv.conf(4) for more details):

```
hosts:                  nisplus dns [NOTFOUND=return] files
```

Enumeration – getXXXent()

Many of the databases have enumeration functions: passwd has getpwent(), hosts has gethostent(), and so on. These were reasonable when the only source was files but often make little sense for hierarchically structured sources that contain large numbers of entries, much less for multiple sources. The interfaces are still provided and the implementations strive to provide reasonable results, but the data returned may be incomplete (enumeration for hosts is simply not supported by the dns source), inconsistent (if multiple sources are used), formatted in an unexpected fashion (for a host with a canonical name and three aliases, the nisplus source will return four hostents, and they may not be consecutive), or very expensive (enumerating a passwd database of 5,000 users is probably a bad idea). Furthermore, multiple threads in the same process using the same reentrant enumeration function (getXXXent_r()) are supported beginning with SunOS 5.3) share the same enumeration position; if they interleave calls, they will enumerate disjoint subsets of the same database.

In general, the use of the enumeration functions is deprecated. In the case of `passwd`, `shadow`, and `group`, it may sometimes be appropriate to use `fgetgrent()`, `fgetpwent()`, and `fgetspent()` (see `getgrnam(3C)`, `getpwnam(3C)`, and `getspnam(3C)`, respectively), which use only the files source.

FILES

A source named `SSS` is implemented by a shared object named `nss_SSS.so.1` that resides in `/usr/lib`.

<code>/etc/nsswitch.conf</code>	Name service configuration file.
<code>/usr/lib/nss_compat.so.1</code>	Implements "compat" source.
<code>/usr/lib/nss_dns.so.1</code>	Implements "dns" source.
<code>/usr/lib/nss_files.so.1</code>	Implements "files" source.
<code>/usr/lib/nss_nis.so.1</code>	Implements "nis" source.
<code>/usr/lib/nss_nisplus.so.1</code>	Implements "nisplus" source.
<code>/etc/netconfig</code>	Configuration file for <code>netdir(3N)</code> functions that redirects hosts/devices policy to the switch.
<code>/etc/nsswitch.files</code>	Sample configuration file that uses "files" only.
<code>/etc/nsswitch.nis</code>	Sample configuration file that uses "files" and "nis".
<code>/etc/nsswitch.nisplus</code>	Sample configuration file that uses "files" and "nisplus".

**SUMMARY
OF TRUSTED
SOLARIS
CHANGES**

The following Trusted Solaris files have been added: `tsolprof`, `tsoluser`, `tnrhdb`, and `tnrhttp`.

In the default Trusted Solaris environment, an administrative role uses the Name Service Switch action in the `System_Admin` folder in the Application Manager to edit the `nsswitch.conf` file. This file should not be edited directly.

The Trusted Solaris environment does not support NIS (YP) or compatibility packages.

SEE ALSO
Trusted Solaris 7
Reference Manual

`automount(1M)`, `ifconfig(1M)`, `rpc.bootparamd(1M)`, `rpc.nisd(1M)`, `sendmail(1M)`, `resolv.conf(4)`

**SunOS 5.7 Reference
Manual**

nis+(1), passwd(1), ethers(3N), getgrnam(3C), gethostbyname(3N),
getnetbyname(3N), getnetgrent(3N), getprotobyname(3N),
getpublickey(3N), getpwnam(3C), getrpcbyname(3N),
getservbyname(3N), getsppnam(3C), netdir(3N), secure_rpc(3N),
netconfig(4)

NOTES

Within each process that uses `nsswitch.conf`, the entire file is read only once; if the file is later changed, the process will continue using the old configuration.

Programs that use the `getXXbyYY()` functions cannot be linked statically since the implementation of these functions requires dynamic linker functionality to access the shared objects `/usr/lib/nss_SSS.so.1` at run-time.

The `compat` source may not be supported in future releases.

Misspelled names of sources and databases will be treated as legitimate names of (most likely nonexistent) sources and databases.

The following functions do *not* use the switch: `fgetgrent(3C)`, `fgetpwent(3C)`, `fgetspent(3C)`, `getpw(3C)`, `putpwent(3C)`, `shadow(4)`.

NAME	priv_desc – Descriptions of defined privileges
SYNOPSIS	<code>#include <tsol/priv.h></code>
DESCRIPTION	<p>Every defined privilege has a manifest constant for use in programs, a name for use in user interfaces, and a description displayed by certain administrative tools. When a process has a privilege in its <i>effective</i> set, that process has the power to bypass security policy and perform the task allowed by that privilege. The manifest constant, name, and description for each privilege defined on this system follows.</p> <hr/> <p>Information labels (ILs) are not supported in Trusted Solaris 7 and later releases. Trusted Solaris software interprets any ILs on communications and files from systems running earlier releases as <code>ADMIN_LOW</code>. Therefore, privileges that allow IL operations are not in effect in Trusted Solaris 7 and later releases.</p> <hr/> <p>Manifest Constant <code>PRIV_FILE_AUDIT</code></p> <p>Name <code>file_audit</code></p> <p>Allows a process to get or set a file's or directory's audit preselection information. The audit preselection information may override the preselection information associated with a process' access to a file or directory. Allows a process to get or set a file's or directory's public object flag. The public object flag may override the successful read/search access preselection information associated with a process' access to a file or directory. Allows a process to write to or modify a file or directory without the file's or directory's audit preselection information or public object flag being cleared.</p> <p>Manifest Constant <code>PRIV_FILE_CHOWN</code></p> <p>Name <code>file_chown</code></p> <p>Allows a process to change a file's owner user ID. Allows a process to change a file's group ID to one other than the process' effective group ID or one of the process' supplemental group IDs.</p> <p>Manifest Constant <code>PRIV_FILE_DAC_EXECUTE</code></p> <p>Name <code>file_dac_execute</code></p> <p>Allows a process to execute an executable file whose permission bits or ACL do not allow the process execute permission.</p> <p>Manifest Constant <code>PRIV_FILE_DAC_READ</code></p> <p>Name <code>file_dac_read</code></p>

	Allows a process to read a file or directory whose permission bits or ACL do not allow the process read permission.
Manifest Constant	PRIV_FILE_DAC_SEARCH
Name	file_dac_search
	Allows a process to search a directory whose permission bits or ACL do not allow the process search permission.
Manifest Constant	PRIV_FILE_DAC_WRITE
Name	file_dac_write
	Allows a process to write a file or directory whose permission bits or ACL do not allow the process write permission.
Manifest Constant	PRIV_FILE_DOWNGRADE_SL
Name	file_downgrade_sl
	Allows a process to set the Sensitivity Label of a file or directory to a Sensitivity Label that does not dominate the existing Sensitivity Label.
Manifest Constant	PRIV_FILE_FILE_LOCK
Name	file_lock
	Allows a process to get accurate lock information for a file lock that it does not hold.
Manifest Constant	PRIV_FILE_MAC_READ
Name	file_mac_read
	Allows a process to read a file or directory whose Sensitivity Label is not dominated by the process' Sensitivity Label. Allows a process to get accurate file attributes of a file or directory whose Sensitivity Label is not dominated by the process' Sensitivity Label. Allows a process, when upgraded directory names are hidden, to get directory entries whose Sensitivity Label is not dominated by the process' Sensitivity Label.
Manifest Constant	PRIV_FILE_MAC_SEARCH
Name	file_mac_search
	Allows a process to search a directory whose Sensitivity Label is not dominated by the process' Sensitivity Label.
Manifest Constant	PRIV_FILE_MAC_WRITE
Name	file_mac_write

Allows a process to write a file or directory whose Sensitivity Label does not dominate the process' Sensitivity Label, or whose Sensitivity Label dominates the process' Clearance.

Manifest Constant PRIV_FILE_OWNER

Name file_owner

Allows a process which is not the owner of a file to modify that file's access and modification times, audit preselection attributes, privileges, or downgrade labels. Allows a process which is not the owner of a directory to modify that directory's access and modification times or downgrade labels. Allows a process which is not the owner of a file or directory to remove or rename a file or directory whose parent directory has the "save text image after execution" (sticky) bit set. Allows a process which is not the owner of a file to mount a "namefs" upon that file. (Does not apply to setting access permission bits or ACLs.)

Manifest Constant PRIV_FILE_SETDAC

Name file_setdac

Allows a process which is not the owner of a file or directory to modify that file's or directory's permission bits or ACL.

Manifest Constant PRIV_FILE_SETID

Name file_setid

Allows a process to change the ownership of a file or write to a file without the set-user-ID and set-group-ID bits being cleared. Allows a process to set the set-group-ID bit on a file whose group is not the process' effective group or one of the process' supplemental groups.

Manifest Constant PRIV_FILE_SETPRIV

Name file_setpriv

Allows a process to set the privilege sets on an executable file that the process owns. Allows a process to write to an executable file without the file's allowed and forced privilege sets being emptied.

Manifest Constant PRIV_FILE_UPGRADE_SL

Name file_upgrade_sl

Allows a process to set the Sensitivity Label of a file or directory to a Sensitivity Label that dominates the existing Sensitivity Label.

Manifest Constant PRIV_IPC_DAC_READ

Name	ipc_dac_read
	Allows a process to read a System V IPC Message Queue, Semaphore Set, or Shared Memory Segment whose permission bits or ACL do not allow the process read permission.
Manifest Constant	PRIV_IPC_DAC_WRITE
Name	ipc_dac_write
	Allows a process to write a System V IPC Message Queue, Semaphore Set, or Shared Memory Segment whose permission bits or ACL do not allow the process write permission.
Manifest Constant	PRIV_IPC_MAC_WRITE
Name	ipc_mac_write
	Allows a process to write a System V IPC Message Queue, Semaphore Set, or Shared Memory Segment whose Sensitivity Label does not dominate the process' Sensitivity Label, or whose Sensitivity Label dominates the process' Clearance.
Manifest Constant	PRIV_IPC_OWNER
Name	ipc_owner
	Allows a process which is not the owner of a System V IPC Message Queue, Semaphore Set, or Shared Memory Segment to remove, change ownership of, or change permission bits or ACL of the Message Queue, Semaphore Set, or Shared Memory Segment.
Manifest Constant	PRIV_NET_BROADCAST
Name	net_broadcast
	Allows a process to send broadcast or multicast packets. Because broadcast packets are delivered to all machines on the local network, they are not labeled.
Manifest Constant	PRIV_NET_DOWNGRADE_SL
Name	net_downgrade_sl
	Allows a process to specify a Sensitivity Label for data being written or to set the network endpoint default Sensitivity Label to an Sensitivity Label which does not dominate the process' Sensitivity Label.
Manifest Constant	PRIV_NET_MAC_READ
Name	net_mac_read

Allows a process to bind to or accept with a multi-level port. Binding to a multi-level port allows the process to read all data sent to that port socket for which there is not a bound single level port that matches the Sensitivity Label of the data. Accepting with a multi-level port allows a process to receive all data sent to that connected port. (There can be no single level connected port for the accept to succeed.) Allows a process to create a multi-level RPC port mapping.

Manifest Constant `PRIV_NET_PRIVADDR`

Name `net_privaddr`

Allows a process to bind to a privileged port number. The privilege port numbers are 1-1023 (the traditional UNIX privileged ports) and 6000-6002 (the XSun server ports). Privileged port numbers include the Internet reserved (well known) port numbers.

Manifest Constant `PRIV_NET_RAWACCESS`

Name `net_rawaccess`

Allows a process to have direct access to the network layer. Direct access to the network layer bypasses network labeling. Auditing is not bypassed.

Manifest Constant `PRIV_NET_REPLY_EQUAL`

Name `net_reply_equal`

Allows a process to reply with the Sensitivity Label of the last packet received rather than its own Sensitivity Label. A combination of `net_mac_read` and `net_reply_equal` allow unmodified programs to successfully receive and reply at all Sensitivity Labels. This privilege exists for unmodified program compatibility and is not used by modified Trusted Solaris programs.

Manifest Constant `PRIV_NET_SETCLR`

Name `net_setclr`

Allows a process to specify a Clearance for data being written or to set the network endpoint default Clearance to a value different from the process' Clearance.

Manifest Constant `PRIV_NET_SETID`

Name `net_setid`

Allows a process to specify an effective user ID, effective group ID, or set of supplemental groups for data being written or to set the network endpoint default effective user ID, effective group ID, or set

of supplemental groups to values different from the process' values. Allows a process which is not the owner of a RPC port mapping to remove the mapping.

Manifest Constant PRIV_NET_SETPRIV

Name net_setpriv

Allows a process to specify the effective privilege set for data being written or to set the network endpoint default effective privilege set to privileges contained in the process' permitted privilege set.

Manifest Constant PRIV_NET_UPGRADE_SL

Name net_upgrade_sl

Allows a process to specify a Sensitivity Label for data being written or to set the network endpoint default Sensitivity Label to a Sensitivity Label which dominates the process' Sensitivity Label.

Manifest Constant PRIV_PROC_AUDIT_APPL

Name proc_audit_appl

Allows a process to generate audit records with an audit event outside the Trusted Solaris TCB event number range. Allows a process to get its own audit preselection information.

Manifest Constant PRIV_PROC_AUDIT_TCB

Name proc_audit_tcb

Allows a process to generate audit records with an audit event in the Trusted Solaris TCB event number range. Allows a process to get its own audit preselection information.

Manifest Constant PRIV_PROC_CHROOT

Name proc_chroot

Allows a process to change its root directory.

Manifest Constant PRIV_PROC_DUMPCORE

Name proc_dumpcore

Allows a TCB process to execute a new program which is set-user-ID, set-group-ID, or permits the use of privilege to have a "core" file created for it when taking the default action for SIGQUIT, SIGILL, SIGTRAP, SIGABRT, SIGEMT, SIGFPE, SIGBUS, SIGSEGV, SIGSYS, SIGXCPU, or SIGXFSZ signals. Allows a TCB process to have a "core" file created for it when taking the default action for SIGQUIT, SIGILL,

SIGTRAP, SIGABRT, SIGEMT, SIGFPE, SIGBUS, SIGSEGV, SIGSYS, SIGXCPU, or SIGXFSZ signals.

Manifest Constant PRIV_PROC_MAC_READ

Name proc_mac_read

Allows a process to read another process whose Sensitivity Label is not dominated by the reading process' Sensitivity Label.

Manifest Constant PRIV_PROC_MAC_WRITE

Name proc_mac_write

Allows a process to write another process whose Sensitivity Label does not dominate the writing process' Sensitivity Label, or whose Sensitivity Label dominates the writing process' Clearance.

Manifest Constant PRIV_PROC_NODELAY

Name proc_nodelay

Allows a process to not be delayed when doing operations that are identified as covert channels.

Manifest Constant PRIV_PROC_OWNER

Name proc_owner

Allows a process to read from and write to another process with a different process owner. Allows a process to bind a process to a CPU with a different process owner.

Manifest Constant PRIV_PROC_SETCLR

Name proc_setclr

Allows a process to set its Clearance to a Clearance that is not equal to the process' current Clearance.

Manifest Constant PRIV_PROC_SETID

Name proc_setid

Allows a process to set its user or group IDs to one different from its current effective, real, or saved IDs. Allows a process to set its supplemental group IDs. Allows a process to set the process group of a controlling terminal to one not in the process' process group. Allows a process to set the window size on a terminal not in its session.

Manifest Constant PRIV_PROC_SETSL

Name proc_setsl

Allows a process to set its Sensitivity Label to a Sensitivity Label that is not equal to the process' current Sensitivity Label.	
Manifest Constant	PRIV_PROC_TRANQUIL
Name	proc_tranquil
Allows a process to set the Sensitivity Label of an object to a Sensitivity Label that is not equal to the current Sensitivity Label when the object is in use by another process.	
Manifest Constant	PRIV_SYS_AUDIT
Name	sys_audit
Allows a process to start the (kernel) audit daemon. Allows a process to view and set the audit state (audit user ID, audit terminal ID, audit session ID, audit preselection mask). Allows a process to turn off and on auditing. Allows a process to configure the audit parameters (cache and queue sizes, event to class mappings, policy options).	
Manifest Constant	PRIV_SYS_BOOT
Name	sys_boot
Allows a process to halt, re-boot, or suspend a Trusted Solaris machine.	
Manifest Constant	PRIV_SYS_CONFIG
Name	sys_config
Allows a process to lock into memory and unlock from memory a memory mapped file or Shared Memory Segment. Allows a process to change the scheduling priority of a process not owned by this process, or increase this process' priority. Allows a process to increase its resource or process limits. Allows a process to set the "save text image after execution" (sticky) bit on executable files. Allows a process to turn on and off accounting. Allows a process to change the machine time of day clock. Allows a process to change the machine high resolution timer clock. Allows a process to reconfigure scheduling classes. Allows a process to create and delete (hard) links to directories. Allows a process to place a processor on-line or off-line. Allows a process to modify kernel driver statistics values.	
Manifest Constant	PRIV_SYS_CONSOLE
Name	sys_console
Allows a process to redirect console output to another device.	
Manifest Constant	PRIV_SYS_DEVICES

Name `sys_devices`

Allows a process to create device special files. Allows a process to use `mknod(2)` to create directory and regular files. Allows a process to revoke all access to a device special file. Allows a process to reassign a controlling terminal from one process to another. Allows a process to open a terminal already exclusively opened. Allows a process to revoke access to its controlling terminal. Allows a process to enable or disable keyboard abort processing. Allows a process to map frame buffer devices into its address space. Allows a process to enable or disable a disk's write-check capability. Allows a process to load a kernel loadable driver. Allows a process to control the Floating Point Accelerator. Allows a process to configure autopush STREAMS modules. Allows a process to configure the device driver policy table. Allows a process to successfully call a third party loadable module that calls the kernel `drv_priv(9F)` function to check for allowed access.

Manifest Constant `PRIV_SYS_FS_CONFIG`

Name `sys_fs_config`

Allows a process to manipulate filesystem locks. Allows a process to set/clear the automatic update (delayed I/O) state of a filesystem. Allows a process to get meta disk allocation information. Allows a process to open a specified inode in a filesystem. Allows a process to set the last access time of a file system object.

Manifest Constant `PRIV_SYS_IPC_CONFIG`

Name `sys_ipc_config`

Allows a process to increase the size of a System V IPC Message Queue buffer.

Manifest Constant `PRIV_SYS_MAXPROC`

Name `sys_maxproc`

Allows a process to create processes when the maximum number of processes for this process' owning user is exceeded. Allows a process to create the last available process in the system.

Manifest Constant `PRIV_SYS_MINFREE`

Name `sys_minfree`

Allows a process to write to a filesystem whose available storage space is below the minimum allowed.

Manifest Constant `PRIV_SYS_MOUNT`

Name `sys_mount`

Allows a process to mount filesystems which are restricted from being freely mounted. Such filesystems include those of type `ufs`, `nfs`, `tmpfs`, `procfs`, ... Allows a process to remount the root filesystem. Allows a process to add and remove swap filesystems. Allows a process to determine the users of a filesystem.

Manifest Constant `PRIV_SYS_NET_CONFIG`

Name `sys_net_config`

Allows a process to configure a machine's network interfaces and routes. Allows a process to set a machine's host and domain names. Allows a process to set a machine's kerberos realm. Allows a process to load and unload host type, accreditation, and default information. Allows a process direct access to network devices. Allows a process to set endpoint names. Allows a process to use the `rpcmod STREAMS` module.

Manifest Constant `PRIV_SYS_NFS`

Name `sys_nfs`

Allows a process to start a kernel NFS daemon. Allows a process to start and stop a kernel NFS lock manager daemon. Allows a process to export directories for use by NFS clients. Allows a process to retrieve the NFS file handle for a path name. Allows a process to revoke NFS RPC credentials for a client it does not own.

Manifest Constant `PRIV_SYS_SUSER_COMPAT`

Name `sys_suser_compat`

Allows a process to successfully call a third party loadable module that calls the kernel `suser()` function to check for allowed access. This privilege exists only for third party loadable module compatibility and is not used by Trusted Solaris.

Manifest Constant `PRIV_SYS_SYSTEM_DOOR`

Name `sys_system_door`

Allows a process to create a door that can be opened by processes at any Sensitivity Label.

Manifest Constant `PRIV_SYS_TRANS_LABEL`

Name `sys_trans_label`

Allows a process to translate labels to and from “external string form” that are not dominated by the process’ Sensitivity Label.

Manifest Constant PRIV_WIN_COLORMAP

Name win_colormap

Allows a process to override colormap restrictions. Allows a process to install or remove colormaps. Allows a process to retrieve colormap cell entries allocated by other processes.

Manifest Constant PRIV_WIN_CONFIG

Name win_config

Allows a process to configure or destroy resources that are permanently retained by the X server. Allows a process to use SetScreenSaver to set the screen saver timeout value. Allows a process to use ChangeHosts to modify the display access control list. Allows a process to use GrabServer. Allows a process to use the SetCloseDownMode request which may retain window, pixmap, colormap, property, cursor, font, or graphic context resources.

Manifest Constant PRIV_WIN_DAC_READ

Name win_dac_read

Allows a process to read from a window resource that it does not own (has a different user ID).

Manifest Constant PRIV_WIN_DAC_WRITE

Name win_dac_write

Allows a process to write to or create a window resource that it does not own (has a different user ID). A newly created window property is created with the window’s user ID.

Manifest Constant PRIV_WIN_DEVICES

Name win_devices

Allows a process to perform operations on window input devices. Allows a process to get and set keyboard and pointer controls. Allows a process to modify pointer button and key mappings.

Manifest Constant PRIV_WIN_DGA

Name win_dga

Allows a process to use the direct graphics access (DGA) X protocol extensions. Direct process access to the frame buffer is still required.

Thus the process must have MAC and DAC privileges that allow access to the frame buffer, or the frame buffer must be allocated to the process.

Manifest Constant PRIV_WIN_DOWNGRADE_SL

Name win_downgrade_sl

Allows a process to set the Sensitivity Label of a window resource to a Sensitivity Label that does not dominate the existing Sensitivity Label.

Manifest Constant PRIV_WIN_FONTPATH

Name win_fontpath

Allows a process to set a font path.

Manifest Constant PRIV_WIN_MAC_READ

Name win_mac_read

Allows a process to read from a window resource whose Sensitivity Label is not equal to the process Sensitivity Label.

Manifest Constant PRIV_WIN_MAC_WRITE

Name win_mac_write

Allows a process to write to create a window resource whose Sensitivity Label is not equal to the process Sensitivity Label. A newly created window property is created with the window's Sensitivity Label.

Manifest Constant PRIV_WIN_SELECTION

Name win_selection

Allows a process to request inter-window data moves without the intervention of the selection arbitrator.

Manifest Constant PRIV_WIN_UPGRADE_SL

Name win_upgrade_sl

Allows a process to set the Sensitivity Label of a window resource to a Sensitivity Label that dominates the existing Sensitivity Label.

FILES

/usr/lib/tsol/locale/locale/priv_name

Privileges descriptions

</usr/include/sys/tsol/priv_names.h>

Manifest constant and ID value definitions

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsu

SEE ALSO

**Trusted Solaris 7
Reference Manual**

`Intro(2)`, `getfpriv(2)`, `setfpriv(2)`, `priv_to_str(3)`,
`set_effective_priv(3)`, `priv_name(4)`, `priv_macros(5)`

Trusted Solaris administrator's document set, Trusted Solaris Developer's Guide

**SunOS 5.7 Reference
Manual**

`attributes(5)`

NAME	priv_name – Privilege description database				
SYNOPSIS	</usr/lib/tsol/locale/ <i>locale</i> /priv_name>				
DESCRIPTION	<p>The <i>priv_name</i> database specifies localized privilege names and descriptions defined on this system. This database is used along with the <sys/tsol/priv_names.h> file by <i>priv_to_str</i>(3), <i>str_to_priv</i>(3), and <i>get_priv_text</i>(3) to translate between privilege ID, privilege name string, and description.</p> <p>Each entry in the <i>priv_name</i> database consists of one line with fields separated by colons (:). A line ending with a backslash (\) indicates continuation of the entry on the next line. Lines beginning with a # character are treated as comments. Each entry has the form:</p> <p><i>constant:name:description</i></p> <p>The entry fields are:</p> <p><i>constant</i> The <i>constant</i> field must be identical to the manifest constant defined for the privilege in the <sys/tsol/priv_names.h> file, where a unique privilege ID is assigned to each privilege constant.</p> <p><i>name</i> The external name of the privilege. It is returned by <i>priv_to_str</i>() and is used by <i>str_to_priv</i>(). It is also used by commands like <i>ppriv</i> and <i>pprivtest</i>. The external name can be customized and localized.</p> <p><i>description</i> The description of the privilege. It is returned by <i>get_priv_text</i>(). The description can be customized and localized.</p>				
ATTRIBUTES	<p>See <i>attributes</i>(5) for descriptions of the following attributes:</p> <table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th><th>ATTRIBUTE VALUE</th></tr> </thead> <tbody> <tr> <td>Availability</td><td>SUNWtsu</td></tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	SUNWtsu
ATTRIBUTE TYPE	ATTRIBUTE VALUE				
Availability	SUNWtsu				
EXAMPLES	<p>EXAMPLE 1 A <i>priv_name</i> entry</p> <pre># Example entry in /usr/lib/tsol/locale/C/priv_name # PRIV_PROC_SETID:proc_setid: Allows a process to set its user or group ID to \ one different from its current effective, real, or saved IDs. \ Allows a process to set its supplemental group IDs. \ Allows a process to set the process group of a controlling terminal \ to one not in the process' process group. \ Allows a process to set the window size on a terminal not in its \ session.</pre>				

SEE ALSO**Trusted Solaris 7
Reference Manual**

priv_to_str(3), priv_desc(4)

**SunOS 5.7 Reference
Manual**

attributes(5)

NAME	resolv.conf – Configuration file for name server routines
DESCRIPTION	<p>This file helps initialize routines from the <code>resolver(3N)</code> C library. The resolver routines provide access to the Internet Domain Name System.</p> <p>The resolver configuration file contains information that is read by the resolver routines the first time a process calls them. The file is designed to be human readable and contains a list of keyword-value pairs that provide various types of resolver information. Keyword-value pairs are of the form:</p> <p><i>keyword value</i></p> <p>The different configuration options are:</p> <p><code>nameserver address</code> Specifies the Internet address in dot-notation format of one name server to which the resolver should direct any queries. Up to <code>MAXNS</code> (currently three) name servers may be listed, on as many as <code>MAXNS</code> <code>nameserver</code> lines in <code>resolv.conf</code>. If multiple servers are specified, the resolver routines query them in the order listed. If no <code>nameserver</code> lines are present in the file, resolver routines use the name server on the local machine.</p> <p>The algorithm of the resolver routines is: try the first name server specified. If the query times out, try the next server listed in the configuration file, and so on until the complement of servers there has been exhausted. If those queries also time out, try the full complement of name servers again, until the maximum number of retry passes has been made.</p> <p><code>domainname</code> Specifies a local domain name for use as the default domain.</p> <p>Most queries for names within a domain can use short names relative to the local domain. If a <code>domain</code> line is missing from the configuration file, the domain is determined from the environment variable, <code>LOCALDOMAIN</code>, if it is defined, from the domain name (see <code>domainname(1M)</code>) by omitting the first level, or from the host name (<code>gethostname(3C)</code>) by using everything after the first dot. Finally, if the</p>

	host name does not contain a domain part, the root domain is assumed.
<code>search</code> <i>searchlist</i>	<p>Specifies a search list for host-name lookup. The search list is normally determined from the local domain name; by default, it contains only the local domain name. This may be changed by listing the desired domains for searches in <i>searchlist</i>. Spaces or tabs must separate domain names.</p> <p>Most resolver queries are attempted using each component of the search path in turn until a match is found. Note that this process may be slow and will generate a lot of network traffic if the servers for the listed domains are not local. Also queries will time out if no server is available for one of the domains.</p> <p>The search list is currently limited to six domains with a total of 256 characters.</p>
<code>sortlist</code> <i>addresslist</i>	<p>Causes addresses returned by <code>gethostbyname(3C)</code> to be sorted in accordance with local rules. A sortlist is specified by IP address netmask pairs. The netmask is optional and defaults to the natural netmask of the net. The IP address and optional network pairs are separated by slashes. Up to 10 pairs may be specified. For example, the following specification requires <code>gethostbyname()</code> to return the netmask pair 130.155.160.0/255.255.240.0 ahead of the IP address 130.155.0.0.</p> <pre>sortlist 130.155.160.0/255.255.240.0 130.155.0.0</pre>
<code>options</code> <i>optionlist</i>	<p>Specifies optional behaviors for various resolver routines in accordance with <i>optionlist</i> values, each of which is equivalent to an internal resolver variable.</p> <p>The values that may be included as individual <i>optionlist</i> values are:</p>

<code>debug</code>	Sets <code>RES_DEBUG</code> in the <code>_res.options</code> field.
<code>ndots:n</code>	Sets a floor threshold for the number of dots which must appear in a name given to <code>res_query()</code> (see <code>resolver(3N)</code>) before an initial absolute (as-is) query is performed. The default for <code>n</code> is 1. Thus, if there are any dots in a name, the name is tried first as an absolute name before any search-list domain names are appended to it.
<code>retry:n</code>	Sets the number of attempts made to connect to each name server. While <code>retry:0</code> is allowed, it is equivalent to <code>retry:1</code> . The default is 4.
<code>retrans:n</code>	Sets the basic retransmit timeout, in seconds. The default is 5. An exponential backoff algorithm is used, so the default values for <code>retry</code> and <code>retrans</code> result in $5+10+20+40=75$ seconds of total timeout for each name server. While <code>retrans:0</code> is allowed, it is equivalent to <code>retrans:1</code> .

The `domain` and `search` keywords are mutually exclusive. If more than one instance of these keywords is present, the last instance takes precedence.

The options established through any `search` lines in the local `resolv.conf` file can be overridden on a per-process basis by setting the environment variable, `LOCALDOMAIN`, to a space-separated list of search domains.

The options established through any `options` lines in the local `resolv.conf` file can be amended on a per-process basis by setting the environment variable, `RES_OPTIONS`, to a space-separated list of resolver options. These options are listed above under the `options` keyword.

The keyword-value pair must appear on a single line, and the keyword (for instance, `nameserver`) must start the line. The value or value list follows the keyword, separated from it by white space characters.

To protect `/etc/resolv.conf` from unauthorized modification, it must have a sensitivity label of `ADMIN_LOW`. The DNS name servers specified in these files can reside on either Trusted Solaris hosts or non-trusted hosts. Administrators are advised to configure only DNS name servers on Trusted Solaris hosts in the `/etc/resolv.conf` file.

`/etc/resolv.conf` must have a sensitivity label of `ADMIN_LOW`.

SUMMARY OF TRUSTED SOLARIS CHANGES

FILES

`/etc/resolv.conf` Configuration file for name server routines.

SEE ALSO

Trusted Solaris 7
Reference Manual

`in.named(1M)`, `resolver(3N)`

SunOS 5.7 Reference
Manual

`gethostbyname(3N)`

Vixie, Paul; Dunlap, Keven J., Karels, Michael J., *Name Server Operations Guide for BIND* (public domain), Internet Software Consortium, 1996.

NAME	rmtab – Remote mounted file system table	
SYNOPSIS	/etc/rmtab	
DESCRIPTION	<p>rmtab contains a table of file systems that are remotely mounted by NFS clients. This file is maintained by mountd(1M), the mount daemon. The data in this file should be obtained only from mountd(1M) using the MOUNTPROC_DUMP remote procedure call.</p> <p>The file contains a line of information for each remotely mounted file system. There are a number of lines of the form:</p> <p style="text-align: center;"><i>hostname: fsname</i></p> <p>The mount daemon adds an entry for any client that successfully executes a mount request and deletes the appropriate entries for an unmount request.</p> <p>Lines beginning with a hash ('#') are commented out. These lines are removed from the file by mountd(1M) when it first starts up. Stale entries may accumulate for clients that crash without sending an unmount request.</p> <p>The /etc/rmtab file must have a sensitivity label of ADMIN_LOW and be owned by UID 0.</p>	
SUMMARY OF TRUSTED SOLARIS CHANGES		
FILES	/etc/rmtab	Remote mounted file system table.
SEE ALSO Trusted Solaris 7 Reference Manual	mountd(1M), showmount(1M)	

NAME	sel_config – Selection rules for copy, cut, paste, drag and drop operations										
SYNOPSIS	/usr/dt/config/sel_config										
DESCRIPTION	<p>The <code>sel_config</code> file specifies how the system behaves when a user performs cut-and-paste, copy-and-paste, and drag-and-drop operations on data between windows that have different sensitivity label. There are two types of entries in this file: automatic confirmation and automatic reply.</p> <p>Automatic Confirmation</p> <p>This type of entry specifies whether a confirmation window (the selection confirmer) displays. Each entry has the form:</p> <p><i>relationship: confirmation</i></p> <p><i>relationship</i> identifies the result of comparing the selected data's source and destination windows' SLs. There are 3 allowed values:</p> <table> <tr> <td>upgradesl</td><td>The source window's sensitivity label is less than the destination window's label.</td></tr> <tr> <td>downgradesl</td><td>The source window's sensitivity label is higher than the destination window's label.</td></tr> <tr> <td>disjointsl</td><td>The source and destination windows' sensitivity labels are disjoint (neither dominates the other).</td></tr> </table> <p><i>confirmation</i> specifies whether to perform automatic confirmation. Allowed values are:</p> <table> <tr> <td>y</td><td>Use automatic confirmation (that is, do not display the selection confirmer window).</td></tr> <tr> <td>n</td><td>Use manual confirmation (that is, display the selection confirmer window). This is the default.</td></tr> </table> <p>Automatic Reply</p> <p>This set of entries provides a means to reduce the number of confirmations that are required of the user, since a single user operation may involve several flows of information between the source and destination windows.</p> <p>There must be one entry of this form:</p> <p><i>autoreply: value</i></p> <p>If <i>value</i> is y (for yes), then the remaining entries of the set are used as attributes for the selection data (rather than the actual contents) to complete the operation without confirmation. If <i>value</i> is n (for no), then the remaining entries are ignored.</p> <p>Defaults can be specified for any <i>type</i> field that appears in the Confirmer window. Below are some examples entries for defaults.</p>	upgradesl	The source window's sensitivity label is less than the destination window's label.	downgradesl	The source window's sensitivity label is higher than the destination window's label.	disjointsl	The source and destination windows' sensitivity labels are disjoint (neither dominates the other).	y	Use automatic confirmation (that is, do not display the selection confirmer window).	n	Use manual confirmation (that is, display the selection confirmer window). This is the default.
upgradesl	The source window's sensitivity label is less than the destination window's label.										
downgradesl	The source window's sensitivity label is higher than the destination window's label.										
disjointsl	The source and destination windows' sensitivity labels are disjoint (neither dominates the other).										
y	Use automatic confirmation (that is, do not display the selection confirmer window).										
n	Use manual confirmation (that is, display the selection confirmer window). This is the default.										

replytype: TARGETS
replytype: Pixel Sets
replytype: LENGTH
replytype: Type Of Monitor

The TARGETS entry, when used, returns the list of target atoms that are supported by the source window. The Pixel Sets and Type Of Monitor entries, are used for animation during a drag-and-drop operation. The LENGTH entry specifies the number of bytes in the selection.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsu

SEE ALSO

Trusted Solaris 7
Reference Manual

Trusted Solaris administrator's document set

SunOS 5.7 Reference
Manual

attributes(5)

NAME	sharetab – Shared file system table											
DESCRIPTION	<p>sharetab resides in directory <code>/etc/dfs</code> and contains a table of local resources shared by the <code>share</code> command.</p> <p>Each line of the file consists of the following fields:</p> <p><i>pathname resource fstype specific_options description</i></p> <p>where</p> <table><tr><td><i>pathname</i></td><td>Indicate the path name of the shared resource.</td></tr><tr><td><i>resource</i></td><td>Indicate the symbolic name by which remote systems can access the resource.</td></tr><tr><td><i>fstype</i></td><td>Indicate the file system type of the shared resource.</td></tr><tr><td><i>specific_options</i></td><td>Indicate filesystem-type-specific options that were given to the <code>share</code> command when the resource was shared.</td></tr><tr><td><i>description</i></td><td>Describe the shared resource provided by the system administrator when the resource was shared.</td></tr></table>		<i>pathname</i>	Indicate the path name of the shared resource.	<i>resource</i>	Indicate the symbolic name by which remote systems can access the resource.	<i>fstype</i>	Indicate the file system type of the shared resource.	<i>specific_options</i>	Indicate filesystem-type-specific options that were given to the <code>share</code> command when the resource was shared.	<i>description</i>	Describe the shared resource provided by the system administrator when the resource was shared.
<i>pathname</i>	Indicate the path name of the shared resource.											
<i>resource</i>	Indicate the symbolic name by which remote systems can access the resource.											
<i>fstype</i>	Indicate the file system type of the shared resource.											
<i>specific_options</i>	Indicate filesystem-type-specific options that were given to the <code>share</code> command when the resource was shared.											
<i>description</i>	Describe the shared resource provided by the system administrator when the resource was shared.											
SUMMARY OF TRUSTED SOLARIS CHANGES	The <code>/etc/dfs/sharetab</code> file must have a sensitivity label of <code>ADMIN_LOW</code> and be owned by <code>UID 0</code> .											
FILES	<code>/etc/dfs/sharetab</code>	Shared file system table.										
SEE ALSO												
Trusted Solaris 7 Reference Manual	<code>share(1M)</code>											

NAME	tndlog – Log of tnd debugging information
SYNOPSIS	/var/tsol/tndlog
DESCRIPTION	<p>/var/tsol/tndlog is the default log file for debugging tnd(1M). This file contains one record for each debugging message. Each record contains the debugging message and time.</p> <p>tndlog is a text file. Each field within each entry is separated from the next by a colon. Each entry is separated from the next by a new line.</p> <p>By default, tndlog does not exist, so no logging is done. To enable logging, tnd must be started with a debug level, or tnctl(1M) must be issued to turn on debugging. The log file can be created either by tnd or by administrators running with appropriate privileges.</p> <p>/var/tsol/tndlog should have a sensitivity label of ADMIN_LOW with permission bits 200, owner root, and group sys.</p>
FILES	/var/tsol/tndlog Log of tnd debugging information
SEE ALSO Trusted Solaris 7 Reference Manual	tnctl(1M), tnd(1M)

NAME	tnidb – Trusted network interface-control database								
SYNOPSIS	<code>/etc/security/tsol/tnidb</code>								
DESCRIPTION	<p>The <code>tnidb</code> database specifies the accreditation range and default security attributes for each network interface. Network traffic is not permitted for any interface, including the loopback interface, unless the interface has a valid entry.</p> <p>Each entry in the interface database consists of one long line, with fields of the entry separated by semicolons (;):</p> <pre><i>interface_name:field1;field2;field3;fieldn;</i></pre> <p>A pound sign (#) as the first character of a line indicates a comment line, which is ignored. Each entry consists of a line of this form:</p> <pre><i>interface_name:min_sl=value;max_sl=value;def_label=value;def_cl=value; def_uid=value;def_gid=value;forced_privs=value;</i></pre> <hr/> <p>The width of this man page prevents showing the foregoing entry on a single line. However, each entry in the database <i>must</i> be a single line.</p> <hr/> <p>The first field for each entry is the interface name. Each entry must contain valid specifications for the accreditation range of the interface for all enforceable security attributes. All fields are mandatory; each entry contains these fields:</p> <table> <tr> <td><code>min_sl, max_sl</code></td><td>Specify the accreditation range of the interface. Only packets with a sensitivity label within the specified accreditation range are allowed into or out of the interface. For a configuration that allows for traffic at all labels, the range should be <code>ADMIN_LOW</code> (in hex) to <code>ADMIN_HIGH</code> (in hex).</td></tr> <tr> <td><code>def_label</code></td><td>Apply this default label to a packet received from an approved remote host that does not support mandatory access control. Under these conditions, all packets imported from the interface that are not labeled with a sensitivity label or information label are assigned this default label. If an information label is not specified, <code>ADMIN_LOW</code> will be used.</td></tr> <tr> <td><code>def_cl</code></td><td>Apply this default clearance to a packet received from an approved remote host that does not support mandatory access control.</td></tr> <tr> <td><code>def_uid, def_gid</code></td><td>Apply this default effective user ID and default effective group ID to a packet. Specify the</td></tr> </table>	<code>min_sl, max_sl</code>	Specify the accreditation range of the interface. Only packets with a sensitivity label within the specified accreditation range are allowed into or out of the interface. For a configuration that allows for traffic at all labels, the range should be <code>ADMIN_LOW</code> (in hex) to <code>ADMIN_HIGH</code> (in hex).	<code>def_label</code>	Apply this default label to a packet received from an approved remote host that does not support mandatory access control. Under these conditions, all packets imported from the interface that are not labeled with a sensitivity label or information label are assigned this default label. If an information label is not specified, <code>ADMIN_LOW</code> will be used.	<code>def_cl</code>	Apply this default clearance to a packet received from an approved remote host that does not support mandatory access control.	<code>def_uid, def_gid</code>	Apply this default effective user ID and default effective group ID to a packet. Specify the
<code>min_sl, max_sl</code>	Specify the accreditation range of the interface. Only packets with a sensitivity label within the specified accreditation range are allowed into or out of the interface. For a configuration that allows for traffic at all labels, the range should be <code>ADMIN_LOW</code> (in hex) to <code>ADMIN_HIGH</code> (in hex).								
<code>def_label</code>	Apply this default label to a packet received from an approved remote host that does not support mandatory access control. Under these conditions, all packets imported from the interface that are not labeled with a sensitivity label or information label are assigned this default label. If an information label is not specified, <code>ADMIN_LOW</code> will be used.								
<code>def_cl</code>	Apply this default clearance to a packet received from an approved remote host that does not support mandatory access control.								
<code>def_uid, def_gid</code>	Apply this default effective user ID and default effective group ID to a packet. Specify the								

def_uid and def_gid fields by using the user ID name and group ID names, respectively.

forced_privs Define the effective privileges to be applied to the incoming packet received from a host that does not support privileges. The format of the privilege set is:

forced_privs=priv[,priv][...]|none|empty|all

where

priv The text string (such as net_mac_read) for privilege.
(forced_privs=net_mac_read)

none Apply no privileges.
(forced_privs=none)

empty Take the default from tnidb.
(forced_privs=empty)

all Apply all privileges.
(forced_privs=all)

The default entries for a particular host specified in the tnrhdb(4) database take precedence over default entries specified for an interface in this database, and the specified values are applied only on incoming packets that do not have any attributes.

All labels are specified in their hex format.

If this database is modified while the network is up, the changes do not take effect until tnctl(1M) updates the interface entries.

Errors in the format of this file can be detected by tnchkdb(1M), which should be run on each database once it has been created or modified. (Refer to the tnchkdb man page for more information.)

/etc/security/tsol/tnidb should have a sensitivity label of ADMIN_LOW with permission bits 444, owner sys, and group sys.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsr

EXAMPLES

EXAMPLE 1 Sample interface entries

For the sake of clarity on this man page, examples are shown using a continuation character (\). In the database file, however, the backslash is not permitted because each entry is made on a single line.

```
#
# Sample interface entries.
#
lo0:min_sl=0x0000000000000000000000000000000000000000000000000000000000000000; \
max_sl=0x7fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff; \
def_label=0x00040c0000000000000000000000000000000000000000000000000000000000; \
def_cl=0x0006000000000000000000000000000000000000000000000000000000000000; \
def_uid=nobody; \
def_gid=nobody; \
forced_privs=None;
# Note that default values are not necessary for lookback interfaces
# because ALL attributes are to accompany the data, and default values
# are only for unlabeled hosts.
#
#
le0:min_sl=0x0000000000000000000000000000000000000000000000000000000000000000; \
max_sl=0x0006000000000000000000000000000000000000000000000000000000000000; \
def_label=0x00040c0000000000000000000000000000000000000000000000000000000000; \
def_cl=0x0006000000000000000000000000000000000000000000000000000000000000; \
def_uid=nobody; \
def_gid=nobody; \
forced_privs=None;
le1:min_sl=0x0000000000000000000000000000000000000000000000000000000000000000; \
max_sl=0x0006000000000000000000000000000000000000000000000000000000000000; \
def_label=[0x00040c0000000000000000000000000000000000000000000000000000000000]; \
def_cl=0x0006000000000000000000000000000000000000000000000000000000000000; \
def_uid=nobody; \
def_gid=nobody; \
forced_privs=None;
```

This sample accreditation range for interfaces 1e0 and 1e1 specifies that only packets with a sensitivity label that dominates ADMIN_LOW and is dominated by TS_NATIONALITY:_CNTRY1/CNTRY2 are allowed into or out of the interface through those interfaces.

Note that interpretations vary by definitions in the `label_encodings(4)` file.

FILES`/etc/security/tsol/tnidb`Trusted network interface-control
database**SEE ALSO****Trusted Solaris 7
Reference Manual**`tnd(1M)`, `tnctl(1M)`, `tnchkdb(1M)`, `tnrhdb(4)`**SunOS 5.7 Reference
Manual**`attributes(5)`**WARNINGS**

For proper functioning, the loopback and primary interface need the `min_sl` to be `ADMIN_LOW` (in hex) and the `max_sl` to be `ADMIN_HIGH` (in hex).

NAME	tnrhdb – Trusted network remote-host database
SYNOPSIS	/etc/security/tsol/tnrhdb
DESCRIPTION	<p>The <code>tnrhdb</code> database specifies which remote-host template to use for each host, including the local host, in the distributed system. <code>tnrhdb</code> works together with the <code>tnrhtp(4)</code> database in allowing the administrator to establish the security and network accreditation attributes for each host. The trusted-network software uses a network “hierarchical fallback” mechanism in looking for a <code>tnrhdb</code> entry for a host. The software looks first for an entry specific to the host; if it does not find one, the software falls back to searching for a matching class-C network entry, and then a class-B entry, a class-A entry, and finally a wildcard entry (IP address 0.0.0.0). In the search for a class-C, a class-B, and a class-A entry, the network environment is assumed to be subnetted on a natural (octet) boundary. (Netmasks will be supported in a future release.) If a host’s IP address cannot be matched to some entry in the <code>tnrhdb</code> database, communication with the host is not permitted.</p> <p>Each entry consists of a line of this form:</p> <pre>IP_address:template_name</pre> <p><i>IP_address</i> This field is the IP address of the host or network that has the security properties specified by the <i>template_name</i> defined in the <code>tnrhtp</code> database. IP addresses are specified in the standard Internet decimal dotted notation. The IP addresses of the hosts and networks should match the IP addresses used for the hosts in the <code>hosts(4)</code> database.</p> <p><i>template_name</i> This value must be a valid template name in the <code>tnrhtp</code> database. See man pages for <code>tnrhtp(4)</code> for information on the security attributes. More than one IP address can use the same template. If this database is modified while the network is up, the changes do not take effect until after <code>tnctl(1M)</code> is used to update the remote-host entries. Administrators are allowed to add new entries and modify existing entries while network is up.</p> <p>Errors in the format of this file can be detected by running <code>tnchkdb</code>, which should be run every time the database is modified or created. Refer to the <code>tnchkdb(1M)</code> man page for more information.</p> <p>/etc/security/tsol/tnrhdb should have a sensitivity label of ADMIN_LOW with permission bits 444, owner <code>sys</code>, and group <code>sys</code>.</p>
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsr

EXAMPLES

EXAMPLE 1 A sample tnrhdb

The following example shows a host that uses template unlabeled1, a host that uses template tsol, a subnet that uses template tsol, a subnet that uses template unlabeled2; and every other host uses the default template specified in the wildcard entry.

```
#
# Assume that templates default, tsol, unlabeled1, and unlabeled2 are
# defined in the tnrhtp(4) database.
#
# first one is the localhost entry
192.110.120.6:tsol
192.110.120.0:tsol
192.110.120.7:unlabeled1
192.110.121.0:unlabeled2
0.0.0.0:default
```

FILES

/etc/security/tsol/tnrhdb Trusted network remote-host database

SEE ALSO

Trusted Solaris 7 Reference Manual

tnd(1M), tnchkdb(1M), tnctl(1M), tnidb(4), tnrhtp(4)

SunOS 5.7 Reference Manual

attributes(5)

WARNINGS

For proper functioning, the primary host name must point to templates that have min_sl = ADMIN_LOW (in hex) and max_sl = ADMIN_HIGH (in hex).

Changing a template while the network is up can change the security view of an undetermined number of hosts.

NOTES

The administrator may wish to make one tnrhdb entry for each host running Trusted Solaris 7 and compatible versions, and make one subnet entry that applies to all unlabeled hosts that have the same security attributes. Then, the administrator may make a separate entry for each host that must be assigned a different set of security attributes.

NAME	tnrhttp – Trusted network remote-host templates								
SYNOPSIS	<code>/etc/security/tsol/tnrhttp</code>								
DESCRIPTION	<p>The <code>tnrhttp</code> database of templates is specified by the administrator for convenience when assigning accreditation and security attributes for each host in the distributed system, including the local host and network. <code>tnrhttp</code> works together with <code>tnrhdb(4)</code>; IP addresses in <code>tnrhdb</code> can be assigned only to templates defined in the <code>tnrhttp</code> database. The administrator should run <code>tnchkdb(1M)</code> to check the syntax after each modification to the <code>tnrhttp</code> database.</p> <p>Each entry in the interface database must be formed as one long line, with fields of the entry separated by semicolons (;):</p> <pre>template_name:field1;field2;field3;fieldn;</pre> <p>A pound sign (#) as the first character of a line indicates a comment line, which is ignored.</p> <p>Each entry consists of a line of this form:</p> <pre>template_name: field_name=value;[field_name= value; ...]</pre> <p>Six types of hosts are currently supported: <code>unlabeled</code>, <code>sun_tsol</code>, <code>ripso</code>, <code>cipso</code>, <code>tsix</code>, and <code>msix</code>.</p> <p>All fields of a particular <i>host_type</i> are mandatory even if no value is set other than <code>none</code>. If this database is modified while the network is up, the changes do not take effect immediately unless <code>tnctl(1M)</code> is used to update the template entries; otherwise, the changes take effect when next polled by the trusted network daemon, <code>tnd(1M)</code>. Administrators are allowed to add new templates and modify attributes of existing templates while the network is up.</p> <p><code>/etc/security/tsol/tnrhttp</code> should be at a sensitivity label of <code>ADMIN_LOW</code> with permission bits 444, owner <code>sys</code>, and group <code>sys</code>.</p> <p>When specifying a name for a template, note that only the first 31 characters of the template name are read and interpreted. These characters must be unique. You can use any printable character in a template name except for field delimiters, new-line, or the comment character.</p> <p>The template for the <code>unlabeled</code> host type has these fields:</p> <table> <tr> <td><i>template_name</i></td><td>Specify a name for the template.</td></tr> <tr> <td><i>host_type</i></td><td><code>unlabeled</code></td></tr> <tr> <td><i>def_label</i>, <i>def_cl</i></td><td></td></tr> <tr> <td><i>def_uid</i>, <i>def_gid</i></td><td></td></tr> </table>	<i>template_name</i>	Specify a name for the template.	<i>host_type</i>	<code>unlabeled</code>	<i>def_label</i> , <i>def_cl</i>		<i>def_uid</i> , <i>def_gid</i>	
<i>template_name</i>	Specify a name for the template.								
<i>host_type</i>	<code>unlabeled</code>								
<i>def_label</i> , <i>def_cl</i>									
<i>def_uid</i> , <i>def_gid</i>									

**Template for
unlabeled Hosts**

Define the default attributes to be applied to incoming data from the remote hosts that do not support these attributes. These defaults override the defaults specified for an interface in the `tnidb(4)` database.

To take defaults from `tnidb(4)`, these `def_*` fields must be set equal to empty; for example, `def_uid=empty`.

The information label is assumed to be `ADMIN_LOW` if the information label for `def_label` is not specified. Obsolete. See NOTES.

`min_sl, max_sl`

Specify the accreditation range for remote host gateways using this template. The format is the same as that in the `tnidb(4)` database. All labels are specified in their hex format.

`forced_privs`

Define the effective privileges to be applied to the incoming packet received from a host that does not support privileges. The format of the privilege set is:

`forced_privs=priv[,priv][...]|none|empty|all`

where

priv The text string (such as `net_mac_read`) for privilege.
(`forced_privs=net_mac_read`)

none Apply no privileges. (`forced_privs=none`)

empty Take the default from `tnidb(4)`. (`forced_privs=empty`)

all Apply all privileges. (`forced_privs=all`)

`def_audit_auid, def_audit_mask, def_audit_termid,
def_audit_asid`

Define the default audit attributes to be applied to incoming data from remote hosts that do not support these attributes.

`def_audit_auid` represents the user's audit ID and can be a positive or negative decimal number.

`def_audit_mask` consists of two 32-bit success and failure masks, specified by an 8-byte hexadecimal number. The bits correspond to audit classes defined in `/etc/security/audit_class`.

`def_audit_termid` consists of a 4-byte port number followed by a 4-byte machine number. These are specified by an 8-byte hexadecimal number.

Template for sun_tsol Hosts

`def_audit_asid` represents the audit session ID and can be a positive or negative decimal number.

Host type `sun_tsol` has these fields:

`template_name` Specify a name for the template.

`host_type` `sun_tsol`

`min_sl, max_sl`

Specify the accreditation range for the remote hosts using this template. The format is the same as that in the `tnidb(4)` database: in hex format.

`allowed_privs`

Limit the effective privilege set for an incoming packet. If a source host associated with this template sends a packet to a destination host, the destination will limit the privilege set of the arrival packet to that specified in this field. The format of the privilege set is:

`allowed_privs=priv[,priv][...]|none|empty|all`

where

`priv` The text string (such as `net_mac_read`) for privilege.
(`allowed_privs=net_mac_read`)

`none` Apply no privileges. (`allowed_privs=none`)

`empty` Take the default from `tnidb(4)`. (`allowed_privs=empty`)

`all` Apply all privileges. (`allowed_privs=all`)

`ip_label`

Provide for IP labeling. These are valid types for `ip_label`:

`none` `ripso` and `cipso` options are not used to label data sent to the host. However, `ripso` and `cipso` security options may be sent to the host if the host is acting as a gateway.

`ripso` For hosts that label their packets with the Revised IP Security Option per RFC 1108. If `ripso` is selected for a host, the `ripso_label` field is required.

`cipso` For hosts that label their packets according to the Common IP Security Options (Tag Type 1 only) as detailed by the Trusted Systems Interoperability Group (TSIG). If `cipso` is selected for a host, the `cipso_doi` field is required.

ripso_label	<p>If <code>ip_label</code> is set to <code>RIPSO</code>, then packets for which the host is the final destination will be labeled with the specified <code>RIPSO</code> label. If the host is configured as a gateway, then the host will be able to route packets with the specified <code>RIPSO</code> label.</p> <p>If <code>ip_label</code> is set to <code>none</code> and <code>ripso_label</code> is set, then the host will be able to forward packets labeled with the specified <code>RIPSO</code> label even though packets addressed to the host will not contain a <code>RIPSO</code> label.</p> <p>Set this field explicitly to <code>empty</code> if no value is to be assigned.</p> <p>These are supported classification level encodings: <code>TOP_SECRET</code>, <code>SECRET</code>, <code>CONFIDENTIAL</code>, <code>UNCLASSIFIED</code> or a hexadecimal representation. These are supported protection authority flags: <code>GENSER</code>, <code>SIOP-ESI</code>, <code>SCI</code>, <code>NSA</code>, <code>DOE</code>, or a hexadecimal representation.</p>
ripso_error	<p>These are the protection authority flags that are used to label ICMP messages generated in response to incoming <code>RIPSO</code>-labeled packets: <code>GENSER</code>, <code>SIOP-ESI</code>, <code>SCI</code>, <code>NSA</code>, <code>DOE</code>, or a hexadecimal representation. The classification level is taken from the <code>ripso_label</code> field. The sender's template is always used when labeling ICMP error messages with <code>RIPSO</code> labels.</p> <p>This field can take multiple values; these must be separated by commas.</p> <p>Set this field explicitly to <code>empty</code> if no value is to be assigned.</p>
cipso_doi	<p>This number is the host's domain of interpretation for <code>CIPSO</code>-labeled packets. The domain of interpretation is a field within the <code>CIPSO</code> security option. If <code>ip_label</code> is set to <code>CIPSO</code>, then packets for which the host is the final destination will be labeled with a <code>CIPSO</code> label containing the specified <code>cipso_doi</code>. If the host is configured as a gateway, then the host will be able to route <code>CIPSO</code>-labeled packets containing the specified <code>cipso_doi</code>. To prevent a gateway from routing <code>CIPSO</code>-labeled packets, set this field to <code>none</code>; to allow a non-gateway machine to send and receive <code>CIPSO</code>-labeled packets, set this field to the appropriate DOI.</p> <p>If <code>ip_label</code> is set to <code>none</code> and <code>cipso_doi</code> is set, then the host will be able to forward <code>CIPSO</code>-labeled packets containing the specified <code>cipso_doi</code> even though packets addressed to the host will not contain a <code>CIPSO</code> label.</p>

**Template for ripso
Hosts**

Set this field explicitly to `empty` if no value is to be assigned.

`def_audit_auid`, `def_audit_mask`, `def_audit_termid`,
`def_audit_asid`

Define the default audit attributes to be applied to incoming data from remote hosts that do not support these attributes.

`def_audit_auid` represents the user's audit ID and can be a positive or negative decimal number.

`def_audit_mask` consists of two 32-bit success and failure masks, specified by an 8-byte hexadecimal number. The bits correspond to audit classes defined in `/etc/security/audit_class`.

`def_audit_termid` consists of a 4-byte port number followed by a 4-byte machine number. These are specified by an 8-byte hexadecimal number.

`def_audit_asid` represents the audit session ID and can be a positive or negative decimal number.

The template for `ripso` host type is for non-TSOL hosts that label packets with the RIPSOL basic security option. This template has these fields:

template_name Specify a name for the template.

`host_type` `ripso`

`ripso_label` These are supported classification level encodings: `TOP_SECRET`, `SECRET`, `CONFIDENTIAL`, `UNCLASSIFIED`, or a hexadecimal representation. These are supported protection authority flags: `GENSER`, `SIOP-ESI`, `SCI`, `NSA`, `DOE`, or a hexadecimal representation.

`ripso_error` These are the protection authority flags that are used to label ICMP messages generated in response to incoming RIPSOL-labeled packets.

This field can take multiple values; these must be separated by commas.

`def_label`, `def_cl`, `def_uid`, `def_gid`

Define the default attributes to be applied to incoming data from the remote hosts that do not support these attributes. These defaults override the defaults specified for an interface in the `tnidb(4)` database.

Set this field explicitly to `empty` if no value is to be assigned.

Default labels are not required for the remote-host entry if there are interface defaults that would be the same for the remote host.

`min_sl`, `max_sl`

Specify the accreditation range for the remote host gateway using this template. The format is the same as that in the `tnidb(4)` database: in hex format.

`forced_privs`

Define the effective privileges to be applied to the incoming packet received from a host that does not support privileges. Having no privileges specified is *not* the same as specifying the word `none`. The format of the privilege set is:

`forced_privs=priv[,priv][...]|none|empty|all`

where

`priv` The text string (such as `net_mac_read`) for privilege.
(`forced_privs=net_mac_read`)

`none` Apply no privileges. (`forced_privs=none`)

`empty` Take the default from `tnidb(4)`. (`forced_privs=empty`)

`all` Apply all privileges. (`forced_privs=all`)

`def_audit_auid, def_audit_mask, def_audit_termid,`
`def_audit_asid`

Define the default audit attributes to be applied to incoming data from remote hosts that do not support these attributes.

`def_audit_auid` represents the user's audit ID and can be a positive or negative decimal number.

`def_audit_mask` consists of two 32-bit success and failure masks, specified by an 8-byte hexadecimal number. The bits correspond to audit classes defined in `/etc/security/audit_class`.

`def_audit_termid` consists of a 4-byte port number followed by a 4-byte machine number. These are specified by an 8-byte hexadecimal number.

`def_audit_asid` represents the audit session ID and can be a positive or negative decimal number.

Template for `cipso` Hosts

The template for `cipso` host type is for hosts that use CIPSO Tag Type 1 to label packets. This template has these fields:

`template_name` Specify a name for the template.

`host_type` `cipso`

cipso_doi This number is the host's domain of interpretation for CIPSO-labeled packets.

def_il Specify the default information label to be applied to incoming data from remote hosts using this template. Obsolete. See NOTES.

min_sl, max_sl
Specify the accreditation range for the remote hosts using this template. The format is the same as that in the `tnidb(4)` database: in hex format..

def_cl, def_uid, def_gid
Define the default attributes to be applied to incoming data from the remote hosts that do not support these attributes. These defaults override the defaults specified for an interface in the `tnidb(4)` database.

To take defaults from `tnidb(4)`, these `def_*` fields must be set equal to empty, for example, `def_uid=empty`.

forced_privs
Defines the effective privileges to be applied to the incoming packet received from a host that does not support privileges. Having no privileges specified is *not* the same as specifying the word `none`. The format of the privilege set is:

`forced_privs=priv[,priv][...]|none|empty|all`

where

priv The text string (such as `net_mac_read`) for privilege.
(`forced_privs=net_mac_read`)

none Apply no privileges. (`forced_privs=none`)

empty Take the default from `tnidb(4)`. (`forced_privs=empty`)

all Apply all privileges. (`forced_privs=all`)

def_audit_audit, def_audit_mask, def_audit_termid, def_audit_asid
Define the default audit attributes to be applied to incoming data from remote hosts that do not support these attributes.

`def_audit_audit` represents the user's audit ID and can be a positive or negative decimal number.

`def_audit_mask` consists of two 32-bit success and failure masks, specified by an 8-byte hexadecimal number. The bits correspond to audit classes defined in `/etc/security/audit_class`.

**Template for tsix
Hosts**

`def_audit_termid` consists of a 4-byte port number followed by a 4-byte machine number. These are specified by an 8-byte hexadecimal number.

`def_audit_asid` represents the audit session ID and can be a positive or negative decimal number.

The template for `tsix` host type is for hosts that use TSIX(RE) 1.1 protocols with token mapping to label packets. This template has these fields:

template_name Specify a name for the template.

`host_type` `tsix`

`min_sl, max_sl`

Specify the accreditation range for the remote hosts using this template.

`host_type`.

All labels are specified in their hex format.

`allowed_privs`

Limit the effective privilege set for an incoming packet. If a source host associated with this template sends a packet to a destination host, the destination will limit the privilege set of the arrival packet to that specified in this field. The format of the privilege set is:

`allowed_privs=priv[,priv][...]|none|empty|all`

where

`priv` The text string (such as `net_mac_read`) for privilege.
(`allowed_privs=net_mac_read`)

`none` Apply no privileges. (`allowed_privs=none`)

`empty` Take the default from `tnidb(4)`. (`allowed_privs=empty`)

`all` Apply all privileges. (`allowed_privs=all`)

`forced_privs`

Define the effective privileges to be applied to the incoming packet received from a host that is not supplying privileges. Having no privileges specified is *not* the same as specifying the word `none`. The format of the privilege set is:

`forced_privs=priv[,priv][...]|none|empty|all`

where

<code>priv</code>	The text string (such as <code>net_mac_read</code>) for privilege. (<code>forced_privs=net_mac_read</code>)
<code>none</code>	Apply no privileges. (<code>forced_privs=none</code>)
<code>empty</code>	Take the default from <code>tnidb(4)</code> . (<code>forced_privs=empty</code>)
<code>all</code>	Apply all privileges. (<code>forced_privs=all</code>)
<code>def_label, def_cl, def_uid, def_gid</code>	<p>Define the default attributes to be applied to incoming data from the remote hosts that are not supplying these attributes. These defaults override the defaults specified for an interface in the <code>tnidb(4)</code> database.</p> <p>If you want to take defaults from <code>tnidb(4)</code>, you must set these <code>def_*</code> fields equal to <code>empty</code>; for example, <code>def_uid=empty</code>;</p> <p>Default labels are not required for the remote-host entry if there are interface defaults that would be the same for the remote host. The information label is assumed to be <code>ADMIN_LOW</code> if the information label is not specified in the <code>def_label</code> field. Obsolete. See NOTES.</p>
<code>ip_label</code>	<p>Provide for IP labeling. These are valid types for <code>ip_label</code>:</p> <p>NONE RIPS0 and CIPS0 options are not used to label data sent to the host. However, RIPS0 and CIPS0 security options may be sent to the host if the host is acting as a gateway.</p> <p>RIPS0 For hosts that label their packets with the Revised IP Security Option per RFC 1108. If RIPS0 is selected for a host, the <code>ripso_label</code> field is required.</p> <p>CIPS0 For hosts that label their packets according to the Common IP Security Options (Tag Type 1 only) as detailed by the Trusted Systems Interoperability Group (TSIG). If CIPS0 is selected for a host, the <code>cipso_doi</code> field is required.</p>
<code>ripso_label</code>	If <code>ip_label</code> is set to RIPS0, then packets for which the host is the final destination will be labeled with the specified RIPS0 label. If the host is configured as a gateway, then the host will be able to route packets with the specified RIPS0 label.
<code>ip_label</code>	If set to <code>NONE</code> and <code>ripso_label</code> is set, then the host will be able to forward packets labeled with the specified RIPS0

	<p>label even though packets addressed to the host will not contain a RIPS0 label.</p> <p>These are supported classification level encodings: TOP_SECRET, SECRET, CONFIDENTIAL, UNCLASSIFIED. These are supported protection authority flags: GENSER, SIOP-ESI, SCI, NSA, DOE.</p>
ripso_error	<p>These are the protection authority flags that are used to label ICMP messages generated in response to incoming RIPS0-labeled packets. These are supported protection authority flags: GENSER, SIOP-ESI, SCI, NSA, DOE. The classification level is taken from the ripso_label field. The sender's template is always used when labeling ICMP error messages with RIPS0 labels.</p> <p>This field can take multiple values; these must be separated by commas.</p> <p>If you do not want to assign a value, you must set this field equal to empty.</p>
cipso_doi	<p>This number is the host's domain of interpretation for CIPSO-labeled packets. The domain of interpretation is a field within the CIPSO security option. If ip_label is set to CIPSO, then packets for which the host is the final destination will be labeled with a CIPSO label containing the specified cipso_doi. If the host is configured as a gateway, then the host will be able to route CIPSO-labeled packets containing the specified cipso_doi. To prevent a gateway from routing CIPSO-labeled packets, set this field to 0; to allow a nongateway machine to send and receive CIPSO-labeled packets, set this field to nonzero.</p> <p>If ip_label is set to NONE and cipso_doi is set, then the host will be able to forward CIPSO-labeled packets containing the specified cipso_doi even though packets addressed to the host will not contain a CIPSO label.</p> <p>If you do not want to assign a value, you must set this field equal to empty.</p>
def_audit_auid, def_audit_mask, def_audit_termid, def_audit_asid	<p>Define the default audit attributes to be applied to incoming data from remote hosts that do not support these attributes.</p>

**Template for msix
Hosts**

`def_audit_auid` represents the user's audit ID and can be a positive or negative decimal number.

`def_audit_mask` consists of two 32-bit success and failure masks, specified by an 8-byte hexadecimal number. The bits correspond to audit classes defined in `/etc/security/audit_class`.

`def_audit_termid` consists of a 4-byte port number followed by a 4-byte machine number. These are specified by an 8-byte hexadecimal number.

`def_audit_asid` represents the audit session ID and can be a positive or negative decimal number.

The template for `msix` host type is for hosts that use `msix` protocol to label packets. For example, the template can be used for interoperating with Trusted Solaris 1.2 hosts. The template has these fields:

template_name Specify a name for the template.

`host_type` `msix`

`min_sl, max_sl`

Specify the accreditation range for the remote hosts using this template. The format is the same as that in the `tnidb(4)` database: in hex format.

`def_label, def_cl, def_uid, def_gid`

Define the default attributes to be applied to incoming data from the remote hosts that do not support these attributes. These defaults override the defaults specified for an interface in the `tnidb(4)` database.

If you want to take defaults from `tnidb(4)`, you must set these `def_*` fields equal to `empty`; for example, `def_uid=empty`;

Default labels are not required for the remote-host entry if there are interface defaults that would be the same for the remote host. The information label is assumed to be `ADMIN_LOW` if the information label is not specified in the `def_label` field. Obsolete. See NOTES.

`def_audit_auid, def_audit_mask, def_audit_termid,`
`def_audit_asid`

Define the default audit attributes to be applied to incoming data from remote hosts that do not support these attributes.

`def_audit_auid` represents the user's audit ID and can be a positive or negative decimal number.

`def_audit_mask` consists of two 32-bit success and failure masks, specified by an 8-byte hexadecimal number. The bits correspond to audit classes defined in `/etc/security/audit_class`.

def_audit_termid consists of a 4-byte port number followed by a 4-byte machine number. These are specified by an 8-byte hexadecimal number.

`def_audit_asid` represents the audit session ID and can be a positive or negative decimal number.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsr

EXAMPLES

EXAMPLE 1 Unlabeled Hosts

For the sake of clarity on this man page, examples are shown using a continuation character (`\`). In the database file, however, the backslash is not permitted because each entry is made on a single line.

The information label is assumed to be ADMIN_LOW if the information label is not specified. Obsolete. See NOTES.

```
#  
# A sample tnrtftp template entry for unlabeled machines  
# or networks. In this example, def_gid is taken from tnidb default.  
#  
unlab:host_type=unlabeled;\n    def_label=0x00000000000000000000000000000000000000000000000000000\n000000000000000000000000000000000000000000000000000000000000000000\n000000000000000000000000[0x000000000000000000000000000000000000000000\n00000000000000000000000000000];\n    def_cl=0x00000000000000000000000000000000000000000000000000000\n00000000000000000000;\n    def_uid=nobody;\n    def_gid=empty;\n        min_sl=0x00000000000000000000000000000000000000000000000000000\n00000000000000000000000000;\n        max_sl=0x7fffffffffffffffffffffffffffffffffffffffffffff\nfffffffffffffffffffffffffff;\n    forced_privs=None;\n        def_audit_auid=3; \\\n        def_audit_mask=0x0000000000000000; \\  
        def_audit_termid=0x0000000000000000; \\  
        def_audit_asid=0;
```

EXAMPLE 2 Sun TSOL Hosts

[illegible]

```
ip_label=none;\
ripso_label=empty;\
ripso_error=empty;\
cipso_doi=empty;
    def_audit_auid=3; \
    def_audit_mask=0x0000000000000000; \
    def_audit_termid=0x0000000000000000; \
    def_audit_asid=0;
```

EXAMPLE 3 Sun TSOL and RIPS0

```
#
# A sample tnrtftp template entry for sun_tsol hosts
# or networks that label packets with the RIPS0 security option.
#
tsol_1:host_type=sun_tsol:\
min_sl=0x0000000000000000000000000000000000000000000000000\
000000000000000000000000;\
max_sl=0x7fffffffffffffffffffffffffffffffffffffffffffffffffff\
ffffffffffffffffffff;\
allowed_privs=all;\
ip_label=ripso;\
ripso_label=top_secret sci;\
ripso_error=genser;\
cipso_doi=empty;
    def_audit_auid=3; \
    def_audit_mask=0x0000000000000000; \
    def_audit_termid=0x0000000000000000; \
    def_audit_asid=0;
```

EXAMPLE 4 Sun TSOL and CIPSO

```
#  
# A sample tnrtcp template entry for sun_tsol hosts  
# or networks that label packets with the CIPSO security option.  
#  
tsol_2:host_type=sun_tsol;\nmin_sl=0x000000000000000000000000000000000000000000000000\n000000000000000000000000;\nmax_sl=0x7fffffffffffffffffffffffffffffffffffffffffffff\nfffffffffffffffffff;\nallowed_privs=all;\nip_label=cipso;\nrpso_label=empty;\nrpso_error=empty;\ncipso_doi=1;\ndef_audit_auid=3; \ndef_audit_mask=0x0000000000000000; \ndef_audit_termid=0x0000000000000000; \ndef_audit_asid=0;
```

EXAMPLE 5 RIPSO Security Option

```
#  
# A sample tnrtftp template entry for ripso hosts  
# or networks that label packets with the RIPSO security option.  
#  
ripso:host_type=ripso;\nripso_label=top_secret sci;\nripso_error=genser;\ndef_label=0x0000000000000000000000000000000000000000000000000000000
```

```
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000[0x000000000000000000000000000000000000000
00000000000000000000000000000000];\
def_cl=0x000000000000000000000000000000000000000000000000000000000
0000000000000000000;\
def_uid=nobody;\
def_gid=nobody;\
min_sl=0x000000000000000000000000000000000000000000000000000000000
0000000000000000000;\
max_sl=0x7fffffffffffffffffffffffffffffffffffffffffffffff;
fffffffffffffffffff;\
forced_privs=empty;
def_audit_auid=3; \
def_audit_mask=0x0000000000000000; \
def_audit_termid=0x0000000000000000; \
def_audit_asid=0;
```

EXAMPLE 6 CIPSO Security Option

```
# A sample tnshhttp template entry for cipso hosts
# or networks that label packets with the CIPSO security option.
#
cipso:host_type=cipso;\
cipso_doi=1;\
min_sl=0x0000000000000000000000000000000000000000000000000\
000000000000000000000000;\
max_sl=0x7fffffffffffffffffffffffffffffffffffffffffffffff\
ffffffffffffffff;\
def_il=0x000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000\
000000000000000000000000;\
def_cl=0x000000000000000000000000000000000000000000000000\
0000000000000000000000;\
def_uid=nobody;\
def_gid=nobody;\
forced_privs=empty;
def_audit_auid=3; \
def_audit_mask=0x0000000000000000; \
def_audit_termid=0x0000000000000000; \
def_audit_asid=0;
```

EXAMPLE 7 TSIX Host

[illegible]

```
def_cl=0x7fffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffff;\
def_uid=nobody;\
def_gid=empty;\
ip_label=ripso;\
ripso_label=secret sci;\
ripso_error=doe;\
cipso_doi=empty;
    def_audit_auid=3; \
    def_audit_mask=0x0000000000000000; \
    def_audit_termid=0x0000000000000000; \
    def_audit_asid=0;
```

EXAMPLE 8 MSIX Hosts

[illegible]

The information label is assumed to be `ADMIN_LOW` if the information label is not specified in the `def_label` field. Obsolete. See NOTES.

FILES	/etc/security/tsol/tnrhttp Trusted network remote-host templates
NOTES	<p>Information labels (ILs) are not supported in Trusted Solaris 7 and later releases. Trusted Solaris software interprets any ILs on communications and files from systems running earlier releases as ADMIN_LOW.</p> <p>Objects still have CMW labels, and CMW labels still include the IL component: IL[SL]; however, the IL component is fixed at ADMIN_LOW.</p> <p>As a result, Trusted Solaris 7 has the following characteristics:</p> <ul style="list-style-type: none"> ■ ILs do not display in window labels; SLs (Sensitivity Labels) display alone within brackets. ■ ILs do not float.

- Setting an IL on an object has no effect.
- Getting an object's IL will always return `ADMIN_LOW`.
- Although certain utilities, library functions, and system calls can manipulate IL strings, the resulting ILs are always `ADMIN_LOW`, and cannot be set on any objects.

SEE ALSO

**Trusted Solaris 7
Reference Manual**

**SunOS 5.7 Reference
Manual**

WARNINGS

`tnchkdb(1M)`, `tnd(1M)`, `tnctl(1M)`, `tnidb(4)`

`attributes(5)`

Changing a template while the network is up can change the security view of an undetermined number of hosts.

Allowing unlabeled hosts onto a Trusted Solaris 7 network is a security risk. In order to avoid compromising the rest of your network, such hosts must be *trusted* in the sense that the administrator is certain that they will not be used to compromise the environment. These hosts should also be physically protected to restrict access to authorized individuals.

If you cannot guarantee the physical security of an unlabeled host, it and similar hosts should be isolated on a separate branch of the network. The gateway to the untrusted hosts must be a type `sun_tsol` host, and its database entries for these untrusted hosts and the interface connected to them must be set to reflect the accreditation of these hosts. This setting allows the gateway to label appropriately all packets received from these hosts and to filter packets bound for them.

NAME	tsolgateways – Static routing configuration file								
SYNOPSIS	<code>/etc/tsolgateways</code>								
DESCRIPTION	<p>The <code>/etc/tsolgateways</code> file is used to configure static routes for a host. At system start up, if <code>/etc/tsolgateways</code> exists, its contents are used to set up static routes. If <code>/etc/tsolgateways</code> does not exist, <code>/etc/defaultrouter</code> is checked. If <code>/etc/defaultrouter</code> exists, its contents are used to set up static routes. If neither <code>/etc/tsolgateways</code> nor <code>/etc/defaultrouter</code> exists, then the host uses dynamic routing. For dynamic routing, if <code>in.rdisc(1M)</code> exists, it is used. If the program file <code>/usr/sbin/in.rdisc</code> does not exist, <code>in.routed(1M)</code> is used.</p> <p>The <code>tsolgateways</code> file differs from the <code>defaultrouter</code> file in several ways. The latter can be used only to specify default gateways along with simple metrics that indicate the hop count to the destination. <code>tsolgateways</code> can be used not only to specify default gateways but also to specify gateways for specific hosts and networks. Host and network routing entries in <code>tsolgateways</code> can be specified with an optional <i>emetric</i> that includes security attributes associated with the route. The <i>emetric</i> is used for trusted routing through the shortest route to a destination through gateways whose security level matches the sensitivity of the data being sent out. The <i>emetric</i> is made up of the simple metric plus additional security routing information (SRI). The SRI includes a sensitivity label range and other optional keywords described below.</p> <p>The format of <code>/etc/tsolgateways</code> is shown below:</p> <pre>default [gateway [args]] [extended_metric] or [net host] destination [gateway [args]] [-m emetric] or [net host] destination [gateway [args]] [metric]</pre> <p>where:</p> <table> <tr> <td><i>destination</i></td><td>Is the IP address of the network.</td></tr> <tr> <td><i>gateway</i></td><td>Is the IP address or hostname of the gateway. If a hostname is used, it must be in the <code>/etc/hosts</code> file. Any destination host(s), network(s), and gateway(s) must be specified with an appropriate host type and template in the local or NIS+ versions of the <code>tnrhdb/tnrhtp</code> databases.</td></tr> <tr> <td><i>metric</i></td><td>Is an integer representing the number of hops to the destination network. This option is supported for backward compatibility.</td></tr> <tr> <td><i>emetric</i></td><td>Combines the metric and the SRI of a route, as described below.</td></tr> </table>	<i>destination</i>	Is the IP address of the network.	<i>gateway</i>	Is the IP address or hostname of the gateway. If a hostname is used, it must be in the <code>/etc/hosts</code> file. Any destination host(s), network(s), and gateway(s) must be specified with an appropriate host type and template in the local or NIS+ versions of the <code>tnrhdb/tnrhtp</code> databases.	<i>metric</i>	Is an integer representing the number of hops to the destination network. This option is supported for backward compatibility.	<i>emetric</i>	Combines the metric and the SRI of a route, as described below.
<i>destination</i>	Is the IP address of the network.								
<i>gateway</i>	Is the IP address or hostname of the gateway. If a hostname is used, it must be in the <code>/etc/hosts</code> file. Any destination host(s), network(s), and gateway(s) must be specified with an appropriate host type and template in the local or NIS+ versions of the <code>tnrhdb/tnrhtp</code> databases.								
<i>metric</i>	Is an integer representing the number of hops to the destination network. This option is supported for backward compatibility.								
<i>emetric</i>	Combines the metric and the SRI of a route, as described below.								

The first form uses the `default` keyword to specify a default gateway through which packets are routed if the destination does not match another route specified in the file. If no default is specified and no match can be found among the host or network entries, the packet is dropped.

The third form uses either the `net` or `host` keywords to set up a route to a specific network or host using a simple metric. This form is obsolete.

The second form is like the third form but it uses the `-m` option to specify the *emetric*. The emetric is specified in the following form (with the single line shown as two for readability):

```
metric= val,min_sl=val,max_sl=val,doi= val,
ripso_label= val,ripso_error=val,ripso_only,cipso_only,msix_only
```

If *val* contains a space, the space must be protected by double quotes around the value.

The keywords to be used for the emetric are described below:

<code>metric=</code>	Specify an integer from 0 to 15 for the number of hops to the destination. Mandatory.
<code>min_sl, max_sl</code>	Specify a sensitivity label in either hexadecimal or string form. Mandatory.
<code>doi=</code>	Specify a nonzero integer corresponding to a CIPSO domain of interpretation. If this keyword is specified, do not specify <code>ripso_label</code> , <code>ripso_error</code> , <code>ripso_only</code> , or <code>msix_only</code> .
<code>ripso_label=</code>	Specify the classification, followed by a space, followed by a list of protection authority flags (PAF) separated by semicolons (;). The classification and the PAF flags can be specified either in hexadecimal or string form. The supported classifications are TOP SECRET, SECRET, CONFIDENTIAL, and UNCLASSIFIED. The PAF flags (also referred to as the Send PAF) are GENSER, SIOP-ESI, SCI, NSA, and DOE. If this keyword is specified, <code>ripso_error</code> is required. If this keyword is specified, do not specify <code>doi</code> , <code>cipso_only</code> , or <code>msix_only</code> .
<code>ripso_error=</code>	Specify a list of protection flags separated by semicolons (;) in either hexadecimal or string form. The supported PAF flags (also referred to as the Return PAF) are GENSER, SIOP-ESI,

	SCI, NSA, and DOE. If this keyword is specified, <code>ripso_label</code> is required. If this keyword is specified, do not specify <code>doi</code> , <code>cipso_only</code> , or <code>msix_only</code> .
<code>ripso_only</code>	Specify without a value. If a <code>SUN_RIPSO</code> gateway is involved in a route, use this keyword to indicate that a route can only forward packets having RIPSO labels. If this keyword is specified, <code>ripso_error</code> and <code>ripso_label</code> are required. If this keyword is specified, do not specify <code>doi</code> , <code>cipso_only</code> or <code>msix_only</code> .
<code>cipso_only</code>	Specify without a value. If a <code>SUN_CIPSO</code> gateway is involved in a route, use this keyword to indicate that a route can only forward packets having CIPSO labels. If this keyword is specified, a <code>doi</code> is required. If this keyword is specified, do not specify <code>ripso_label</code> , <code>ripso_error</code> , <code>ripso_only</code> or <code>msix_only</code> .
<code>msix_only</code>	Specify without a value. If a <code>SUN_MSIX</code> gateway is involved in a route, use this keyword to indicate that a route can only forward packets having MSIX labels. If this keyword is specified, do not specify <code>doi</code> , <code>ripso_label</code> , <code>ripso_error</code> , <code>ripso_only</code> or <code>cipso_only</code> .

EXAMPLES

The first two lines in the following example show a default and a network entry, each with a simple metric. The third line shows an entry for a network that specifies the gateway name as `chastain-118`, and the metric as 2, and that assigns an SRI that specifies a label range from UNCLASSIFIED to CONFIDENTIAL, a ripso label of CONFIDENTIAL GENSER, and a ripso error of GENSER. The fourth line is an entry for a host, with an IP address `126.180.101.3`. The host entry specifies a gateway called `trusted`, with a label range of TOP SECRET to TOP SECRET, a `cipso doi` of 1, and the optional keyword `cipso_only`. (The long lines are broken because they do not fit on a single line.)

EXAMPLE 1 Sample `tsolgateways` file

```
default 126.180.117.1 1
net 126.180.113.0 chastain 1
net 126.180.116.0 chastain-118 -m metric=2,min_sl="UNCLASSIFIED",
max_sl="CONFIDENTIAL",ripso_label="CONFIDENTIAL GENSER",
ripso_error="GENSER"
host 126.180.101.3 trusted -m metric=3,min_sl="TOP SECRET",max_sl="TOP SECRET",
doi=1,cipso_only
```

SEE ALSO

**Trusted Solaris 7
Reference Manual**

`in.rdisc(1M)`, `in.routed(1M)`, `route(1M)`, *Trusted Solaris Administrator's Procedures*

NAME	tsolinfo – Package security-attribute description file
DESCRIPTION	<p>tsolinfo describes security attributes used as overrides for file attributes of files contained in a package. This text file is created by the developer of a software package and is included in the package. If the file is not included in the package, a set of default filesystem security attributes will be used.</p> <p>Each entry in the tsolinfo file describes a single file security attribute for a specific file. The entry consists of several fields of information, each field separated by a space. Lines that begin with # are comment lines and are ignored. Empty lines are not allowed. The fields are described below and must appear in the order shown.</p> <p><i>attribute</i> A character field that defines the attribute type. Valid attribute types are:</p> <ul style="list-style-type: none"> label A CMW label in text. The exact label name must be used. See EXAMPLES below. acl A comma-separated list of acl entries terminated with a comma. allowed_privs A list of comma-separated allowed privileges. forced_privs A list of comma-separated forced privileges. mld Specifies a multilevel directory. Do not set an attribute value for this type. public Specifies that read operations on this file should not be audited. Do not set an attribute value for this type. <p><i>pathname</i> A character file that defines the name of the file for which the attribute is being defined.</p> <p><i>attribute-value</i> A character string that defines the value of the attribute. This field is not valid for the mld or public attributes.</p> <p>The tsolinfo file also provides a special set of entries to define a set of default security attributes associated with all of the files within a package. The default attribute is used to denote a default attribute entry. The pathname component of the entry is replaced with the name of the attribute for which the default is being set. Package defaults can be set for any of the attributes described above. The package defaults override the filesystem default security attributes.</p>

The `tsolinfo` file should be created at the same time as the package prototype file is created, and should be located in the same directory. The `tsolinfo` file must be included in the package prototype file by using the package prototype `include` command.

When the `pkgmk(1)` command is used to create a package, the `tsolinfo` file is relocated to the `install/` subdirectory of the newly created package directory.

EXAMPLES

EXAMPLE 1 A sample `tsolinfo` file

```
default label [ADMIN_LOW]
default allowed_privs all
default forced_privs all
label usr/sbin/myfile [ADMIN_HIGH]
forced_privs usr/sbin/myfile file_mac_read
allowed_privs usr/sbin/myfile file_mac_read,file_mac_write
```

EXAMPLE 2 A `tsolinfo` file with an exact CMW label

If an initial compartment is specified for the classification `NEED TO KNOW` and assigned to default word `SSE` in the `SENSITIVITY LABELS: WORDS:` section of the `label_encodings` file, as in:

```
-----
CLASSIFICATIONS:

name= NEED TO KNOW;          sname=NTK;  value= 5; initial compartments= 14;
. . .
SENSITIVITY LABELS:
WORDS:

name= SSE;                   compartments= 14;
-----
```

it is not enough to enter `NEED TO KNOW` in the `tsolinfo` file. The label must include all label components, `NEED TO KNOW SSE`.

```
default label [ADMIN_LOW]
default allowed_privs file_mac_read,file_mac_write
default forced_privs file_mac_read
label usr/sbin/myfile [NEED TO KNOW SSE]
forced_privs usr/sbin/newfile file_mac_read
allowed_privs usr/sbin/newfile file_mac_read,file_mac_write
```

SEE ALSO

Trusted Solaris 7
Reference Manual

SunOS 5.7 Reference
Manual

`setfsattr(1M)`

`pkginfo(4)`, `pkgmap(4)`, `pkgmk(1)`, `prototype(4)`

NOTES

The `tsolinfo` file should only contain entries for pathnames that require special file security attributes, other than the default ones supplied by the UFS filesystem. If the package does not contain any files that require special file security attributes, the `tsolinfo` file should not be created.

If the `tsolinfo` file is not present during package installation, the files contained within a package are assigned default file security attributes provided by the UFS filesystem.

If the `tsolinfo` file contains only the default entries, all of the files within a package are installed with security attributes specified by the `tsolinfo` file entries, along with any non-conflicting default UFS attributes.

NAME	tsolprof – Trusted Solaris User Profiles Database														
SYNOPSIS	/etc/security/tsol/tsolprof														
DESCRIPTION	<p>The <code>tsolprof</code> database specifies security attributes associated with profiles. A profile is a logical grouping of authorizations, commands, and actions, which is interpreted by <code>dtwm(1)</code> and <code>pfsh(1M)</code> to form a secure execution environment. Each user or role account is assigned zero or more profiles in the <code>tsoluser(4)</code> database file.</p> <p><code>tsolprof</code> can be used with other profile sources, namely the NIS+ table <code>tsolprof</code>. Programs use the <code>getprofent(3)</code> routines to gain access to this information.</p> <p>Administrators can modify <code>tsolprof</code> and should do so only through the Profile Manager application, which is accessible through <code>solstice(1M)</code>. Editing the file directly, such as through a text editor, is strongly discouraged.</p> <p>Each entry in the <code>tsolprof</code> database consists of one line of text, containing at least these five elements: the profile name, a short description of its use, a list of authorizations, a list of permitted actions, and a list of permitted commands. Line continuations and comments are not allowed. The basic format of each entry is:</p> <pre>profile:description:authorizations:actions:commands:links:flags</pre> <p>The fields of each entry are:</p> <table><tr><td><code>profile</code></td><td>The name of the profile. If this field contains a caret (^), the entry is interpreted as a continuation of another profile entry.</td></tr><tr><td><code>description</code></td><td>Descriptive text. The field should explain why a user might be assigned the profile.</td></tr><tr><td><code>authorizations</code></td><td>A comma-separated list of authorization numbers or names; or the keyword <code>all</code> or <code>none</code>.</td></tr><tr><td><code>actions</code></td><td>Zero or more semicolon-separated sets of action information; or the key word <code>none</code>, which indicates that execution of actions is not permitted. A set of action information is specified in the form:</td></tr><tr><td></td><td><code>actname;argclass;argtype;argmode;argcount;privs;euid;egid;min;max[;...]</code></td></tr><tr><td><code>actname</code></td><td>Is the name of the action as defined by CDE. This field also accepts the asterisk (*), indicating that all actions executed</td></tr></table>			<code>profile</code>	The name of the profile. If this field contains a caret (^), the entry is interpreted as a continuation of another profile entry.	<code>description</code>	Descriptive text. The field should explain why a user might be assigned the profile.	<code>authorizations</code>	A comma-separated list of authorization numbers or names; or the keyword <code>all</code> or <code>none</code> .	<code>actions</code>	Zero or more semicolon-separated sets of action information; or the key word <code>none</code> , which indicates that execution of actions is not permitted. A set of action information is specified in the form:		<code>actname;argclass;argtype;argmode;argcount;privs;euid;egid;min;max[;...]</code>	<code>actname</code>	Is the name of the action as defined by CDE. This field also accepts the asterisk (*), indicating that all actions executed
<code>profile</code>	The name of the profile. If this field contains a caret (^), the entry is interpreted as a continuation of another profile entry.														
<code>description</code>	Descriptive text. The field should explain why a user might be assigned the profile.														
<code>authorizations</code>	A comma-separated list of authorization numbers or names; or the keyword <code>all</code> or <code>none</code> .														
<code>actions</code>	Zero or more semicolon-separated sets of action information; or the key word <code>none</code> , which indicates that execution of actions is not permitted. A set of action information is specified in the form:														
	<code>actname;argclass;argtype;argmode;argcount;privs;euid;egid;min;max[;...]</code>														
<code>actname</code>	Is the name of the action as defined by CDE. This field also accepts the asterisk (*), indicating that all actions executed														

	will gain any specified privileges, UID, and GID, and be restricted by the given label range.
	When the asterisk is used, the next four fields (<i>argclass</i> , <i>argtype</i> , <i>argmode</i> , and <i>argcount</i>) are irrelevant.
<i>argclass</i>	Is the argument class (for example, <code>FILE</code> or <code>SESSION</code> .)
<i>min</i>	Specifies the minimum label at which the user must operate in order to execute the action.
<i>max</i>	Specifies the maximum label at which the user must operate in order to execute the action.
	To specify another action, place a semicolon after the current <i>actions</i> field and repeat the required information for the next action.
<i>commands</i>	Zero or more semicolon-separated sets of command information; or the keyword <code>none</code> , indicating that no commands are allowed. Each set of command information is specified in the form: <code>dir;filename;privs;euid;egid;min;max[:...]</code>
<i>dir</i>	Is an absolute path to the directory containing the subsequent file names; or the at symbol (<code>@</code>), indicating that the entry applies to the previous <i>dir</i> ; or an asterisk (<code>*</code>), indicating that all commands executed should have the privileges, UID, and GID specified, and should be restricted by the given label range. The first <i>dir</i> entry must be an absolute path. When the asterisk is used, the next field (<i>filename</i>) is not interpreted.
<i>filename</i>	Is the name of the file to which the subsequent attributes apply when the file is executed.

<i>privs</i>	Is a comma-separated list of privilege numbers that make up the effective privileges when the command is executed. The keyword <code>all</code> selects all privileges.
<i>eid, egid</i>	Are similar to the <code>setuid</code> and <code>setgid</code> bits on a file.
<i>min</i>	Specifies the minimum label at which the user must operate in order to execute the action.
<i>max</i>	Specifies the maximum label at which the user must operate in order to execute the action.
<code>links</code>	This field specifies how many profile entries the profile has (in most cases, 1). This field is used to create a profile with more than 8000 bytes of data.
<code>flags</code>	Reserved for future use. It contains the keyword <code>none</code> .

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsr

EXAMPLES

For the sake of clarity on this man page, examples are shown using a continuation character (`\`). In the database file, however, the backslash is not permitted because each entry is made on a single line.

```
All:Foundation. A Standard Solaris User.:none:\
*;*;*;*;:::1:none
Custom Admin Role:Modify this profile to customize the admin\
role for your site.:none:TrustedEditor;*;*;*;4,5,6,43,44\
:::/usr/dt/bin;trusted_edit;4,5,6,43,44;:::1:none
```

The All profile is noted as a "Foundation" profile, meaning that it is best used as the first profile assigned to a user. It happens to provide access to all commands and actions, but without any privileges, label restrictions, or the ability to set the UID or GID of the executed command or action.

The Custom Admin Role profile grants a user the `dac_read` (privilege number 4), `dac_search` (5), `dac_write` (6), `proc_audit_appl` (43), and `proc_audit_tcb` (44) privileges. With this profile a user can edit files. Both the `TrustedEditor` CDE action and the `trusted_edit` CDE command are assigned enough privilege to get past any discretionary file access restrictions

and to write to the audit trail, which is necessary to record changes made to administrative files.

FILES

/etc/security/tsol/tsolprof

Trusted Solaris user profiles
database

WARNINGS

The maximum length of a single line in the profile database is `TSOLPROF_MAX_NIS_ENT` characters as defined in `/etc/default/libtsolddb`. The Profile Manager and the interfaces described in `getprofent(3)` are designed to transparently prevent an entry from exceeding this limit. If the limit is exceeded and the `tsolprof` database is loaded into NIS+, the `rpc.nisd(1M)` may terminate and dump core.

Do not use the following symbols within a profile field: colon (:), semicolon (;), comma (,), caret (^), tab (\t), carriage return (\n), pound (#), or backslash (\).

SEE ALSO

Trusted Solaris 7
Reference Manual

`getprofent(3)`, `auth_name(4)`, `priv_name(4)`, `tsoluser(4)`

SunOS 5.7 Reference
Manual

`solstice(1M)`, `attributes(5)`

NAME	tsoluser – Trusted Solaris User Security Attributes Database
SYNOPSIS	/etc/security/tsol/tsoluser
DESCRIPTION	<p>The <code>tsoluser</code> database specifies additional security attributes associated with users and roles.</p> <p>Each entry in the file consists of a single line, with fields separated by a colon (:). Line continuations and comments are not allowed. Each entry has the form (for readability, the line is shown in this man page as two lines):</p> <pre>user:lock:gen:profiles:roles:idletime:idlecmd:labelview:\ labeltrans:labelmin:clearance:usertype:res1:res2:res3</pre> <p><i>user</i> is the name of the user as specified in the <code>passwd</code> database.</p> <p><i>lock</i> contains one of the keywords: <code>locked</code>, or <code>open</code>. <code>locked</code> specifies that the user is not allowed to log in to the system. <code>open</code> specifies that the user is allowed to log in. Programs such as <code>login(1)</code> and <code>dtlogin(1X)</code> may choose to change the keyword <code>open</code> to <code>locked</code>, for example, when a user enters an invalid password too frequently.</p> <p><i>gen</i> contains either of the strings: <code>automatic</code>, or <code>manual</code>. <code>automatic</code> specifies that a user must choose a machine-generated password to change a password. <code>manual</code> specifies that a user may devise a password of his or her choice.</p> <p><i>profiles</i> contains a ordered, comma-separated list of profile names chosen from <code>tsolprof(4)</code>; or the key word <code>none</code>, indicating that <code>dtwm(1)</code> will not permit the user to use any actions and <code>pfsh(1M)</code> will not permit the user to execute commands.</p> <p><i>roles</i> contains a comma-separated list of role names from the set of user accounts in this database whose <code>usertype</code> field indicates the account is a role; or the keyword <code>none</code>, indicating that the user is not permitted to assume any role.</p> <p><i>idletime</i> contains a number representing the number of minutes a workstation may remain idle before the the window manager attempts the task specified in <i>idlecmd</i>. A zero in this field specifies that the <i>idlecmd</i> command is never executed.</p> <p><i>idlecmd</i> contains one of two keywords which <code>dtwm(1)</code> interprets when a workstation is idle for too long. The keyword <code>lock</code> specifies that the workstation is to be locked (and thus requires the user to provide a password to resume the session). The keyword <code>logout</code> specifies that session is to be terminated (thus killing the user's processes launched in the current session).</p>

labelview contains three comma-separated keywords. The first word can be either *internal*, specifying that the user may see the `ADMIN_LOW` and `ADMIN_HIGH` labels displayed by various commands and applications; or *external*, specifying that the user may not see the labels; or *sysdef*, indicating that the label visibility should be determined by the system default, as recorded in the `label_encodings` file. The second word may be either *show1*, indicating that information labels are not displayed; or *hide1*, indicating that the labels are not displayed. The third word may be *shows1*, indicating that sensitivity labels are displayed; or *hides1*, indicating that the labels are not displayed.

labeltrans contains a hexadecimal number representing the process attribute flags that control label translation.

labelmin contains the minimum sensitivity label at which the user may log in. This label is given as hexadecimal string. See `atohexlabel(1M)`.

clearance contains the maximum sensitivity label at which the user may operate. This label is given as hexadecimal string. See `atohexlabel(1M)`.

usertype contains one of these strings: *utnorm*, indicating that this account is for a normal user, one who logs in; *utrole*, indicating that this account is for a role, which can be assumed by a normal user who is allowed the role after the user has logged in; or *utadm*, indicating that this account is an administrative role with administrative capabilities.

res1, res2, and res3 are reserved for future use.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtSr

EXAMPLES

For the sake of clarity on this man page, examples are shown using a continuation character (\). In the database file, however, the backslash is not permitted because each entry is made on a single line.

[illegible]

```
secadmin:open:automatic:Audit Control,Object Label Management,\
Object Access Management,Object Privilege Management,Outside Accred,\
System Security,NIS+ Security Administration,User Security,\
Basic Commands,Basic Actions:none:5:lock:internal,showil,shows1:0x0000:\
0x0000000000000000000000000000000000000000000000000000000000000000:\
0x7fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff:\
utadm:res1:res2:res3
oper:open:automatic:Outside Accred,Media Backup,Basic Commands,\
Basic Actions:none:5:lock:internal,showil,shows1:0x0000:\
0x0000000000000000000000000000000000000000000000000000000000000000:\
0x7fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff:\
utrole:res1:res2:res3
install:open>manual:Enable Login,Outside Accred:root:5:lock:\
internal,showil,shows1:0x0000:\
0x0000000000000000000000000000000000000000000000000000000000000000:\
0x0000000000000000000000000000000000000000000000000000000000000000:\
utnorm:res1:res2:res3
```

The first four entries (root, admin, secadmin, and oper) are the default Trusted Solaris roles. The first entry, root, has a handful of profiles which allow it to install software and bootstrap the system. Note that the root entry also has the profile named "All," which gives the role wide birth as far as which commands it can execute while installing software.

Each of the next three has a much more restrictive set of profiles. These profiles are designed to allow the user to perform only the tasks for which that role is responsible.

The fifth entry is for the user named install, who has the "Nothing" profile and thus cannot execute any commands or actions. The user can, however, assume the root role, which provides sufficient capability to configure the system immediately after it has been installed.

FILES

/etc/security/tsol/tsoluser Trusted Solaris user security
 attributes database

SEE ALSO

Trusted Solaris 7
Reference Manual

getprofent(3), getuserent(3), pfsh(1M), tsolprof(4)

SunOS 5.7 Reference
Manual

dtwm(1), attributes(5)

NAME	vfstab – Table of file system defaults																					
DESCRIPTION	The file <code>/etc/vfstab</code> describes defaults for each file system. The information is stored in a table with the following column headings:																					
	<table><tr><td>device</td><td>device</td><td>mount</td><td>FS</td><td>fsck</td><td>mount</td><td>mount</td></tr><tr><td>to</td><td>to fsck</td><td>point</td><td>type</td><td>pass</td><td>at boot</td><td>options</td></tr><tr><td>mount</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	device	device	mount	FS	fsck	mount	mount	to	to fsck	point	type	pass	at boot	options	mount						
	device	device	mount	FS	fsck	mount	mount															
	to	to fsck	point	type	pass	at boot	options															
	mount																					
The fields in the table are space-separated and show the resource name (device to mount), the raw device to fsck (device to fsck), the default mount directory (mount point), the name of the file system type (FS type), the number used by fsck to decide whether to check the file system automatically (fsck pass), whether the file system should be mounted automatically by mountall (mount at boot), and the file system mount options (mount options). (See respective mount file system man page below in SEE ALSO for mount options.) A - is used to indicate no entry in a field. This may be used when a field does not apply to the resource being mounted.																						
The <code>getvfsent(3C)</code> family of routines is used to read and write to <code>/etc/vfstab</code> .																						
	<code>/etc/vfstab</code> may be used to specify swap areas. An entry so specified, (which can be a file or a device), will automatically be added as a swap area by the <code>/sbin/swapadd</code> script when the system boots. To specify a swap area, the <i>device-to-mount</i> field contains the name of the swap file or device, the <i>FS-type</i> is "swap", <i>mount-at-boot</i> is "no" and all other fields have no entry.																					
	Mount-time security attributes for a file system specified in the <code>vfstab</code> file can be specified either with the <code>-S</code> option on the <code>mount(1M)</code> command line or in an entry created for the file system in the <code>vfstab_adjunct(4)</code> file. See the DESCRIPTION sections in the <code>mount</code> and the <code>vfstab_adjunct</code> man pages for more about specifying security attributes. The <code>vfstab</code> file should not be edited directly; instead, it should be edited using the Set Mount Points action, which maintains the proper user, group, sensitivity label, and file permissions for the file and audits all changes. The Set Mount Points action resides in the System_Admin folder available in the Application Manager folder in the Front Panel. By default, the administrator (admin) role has the Set Mount Points action in the File System Management execution profile.																					
SUMMARY OF TRUSTED SOLARIS CHANGES	Two new pairs of security-relevant mount options <code>devices nodevices</code> , and <code>priv nopriv</code> can be specified in the <code>vfstab</code> file for filesystems that support them as filesystem-specific options: <code>mount_hfs(1M)</code> , <code>mount_nfs(1M)</code> , and <code>mount_ufs(1M)</code> . Mount-time security attributes can be specified for file systems																					

whose objects do not have any attributes (such as user and group IDs) and for file systems that do not have the Trusted Solaris extended security attributes (such as sensitivity labels). Trusted Solaris security policy applies when mounting. The `vfstab` file should be edited by using the Set Mount Points action.

SEE ALSO

**Trusted Solaris 7
Reference Manual**

**SunOS 5.7 Reference
Manual**

`mount(1M)`, `mount_hsfes(1M)`, `mount_nfs(1M)`, `mount_tmpfs(1M)`,
`mount_ufs(1M)`, `setmnt(1M)`, `vfstab_adjunct(4)`

`fsck(1M)`, `mount_cachefs(1M)`, `swap(1M)`, `getvfsent(3C)`

NAME	vfstab_adjunct – Attribute data file for mounting a file system
SYNOPSIS	/etc/security/tsol/vfstab_adjunct
DESCRIPTION	<p>The <code>vfstab_adjunct</code> file can be used to assign any or all of the following mount-time security attributes to the named file system when appropriate: an ACL, a mode, a user ID, a group ID, a sensitivity label, forced privilege(s), allowed privilege(s), a file attribute flag, a filesystem label range, or an MLD prefix. If the <code>mount(1M)</code> command is called with the <code>-S</code> option to specify security attributes, the <code>vfstab_adjunct</code> file is not consulted. Specifying mount-time attributes is meaningful only when mounting file systems that do not support the attributes.</p> <p>If the file system already has security attributes, the attributes specified at mount time are ignored and a message is issued. Otherwise, the mount time attributes are used as the filesystem-wide security attributes. When access control decisions are made, any security attributes on a file or directory always take precedence over security attributes specified either at the filesystem level or mount time.</p> <p>The <code>vfstab_adjunct</code> file should not be edited directly; instead, it should be edited using the Set Mount Attributes action, which maintains the proper user, group, sensitivity label, and file permissions for the file and audits all changes. The Set Mount Attributes action resides in the <code>System_Admin</code> folder available in the Application Manager folder in the Front Panel. By default, the security administrator (<code>secadmin</code>) role has the Set Mount Attributes action in the File System Security execution profile.</p> <p>Mount-time security attributes can be specified for file systems whose objects do not have any attributes (such as user and group IDs) and for file systems that do not have the Trusted Solaris extended security attributes (such as sensitivity labels). When an appropriate attribute is not specified at mount time for a fixed attribute file system, a default value is applied. The default values are described later in this section.</p> <p>File system types <code>UFS</code>, <code>TMPFS</code>, and <code>NFS</code> (from a Trusted Solaris server) have a full set of Trusted Solaris extended security attributes already defined. (See the <code>getfsattr(1M)</code> man page for how to get attributes on mounted file systems). Because the attributes can be changed on these file systems <i>after</i> they are mounted, they are called <i>variable</i> file systems. For example, the sensitivity label on a file in a variable file system can be changed by an authorized user. Security attributes on variable file systems can be overridden at mount-time, but objects in the file system that have assigned security attributes retain those attributes.</p> <p>File systems that do not support the Trusted Solaris extended security attributes are called <i>fixed</i> because any attributes assigned to them (either at mount time or by default) cannot be changed. For example, the sensitivity label specified for a mounted fixed-attribute file system cannot be changed on any of the objects in</p>

that file system. An object that is moved or copied from the fixed file system to a variable file system can be changed after the move.

Mount-time security attributes override existing security attributes on a file system. However, mount-time attributes never override security attributes on the files and directories within the file system.

Each record in the `vfstab_adjunct` file represents a single file system. An entry consists of the file system's full pathname followed by a semicolon, followed by keyword=value assignments in semicolon-separated fields.

The pathname of the file system is the only portion of the entry that is required and therefore has no keyword associated with it. All keyword fields are optional and follow the format: keyword=value where *keyword* is one of the following:

<code>acc_acl</code>	Sets the same ACL on all files or directories in the file system. See <code>aclfromtext(3)</code> for the format.
<code>mode</code>	<p>Sets a DAC permission mode for each object in the file system. The only supported mode is the absolute mode, which is specified using octal numbers. See the description for the absolute-mode parameter on the <code>chmod(1)</code> man page.</p> <p>(Because the mode is an object-level attribute that has precedence over any mount-time attributes, setting a mode is only meaningful in the rare case when the type of file system being mounted does not support permission bits. In such cases, it is recommended that a value be explicitly specified for the mode.)</p>
<code>attr_flg</code>	<p>Sets an attribute flag on all files in the file system. The only supported <code>attr_flg</code> value is <code>public</code>, whose effect is that when certain read operations are performed on any object in the file system on which this flag is set, audit records are not generated even when the operations are part of a preselected audit class, with the following exception. If the audit pseudo-event for use of privilege (<code>AUE_UPRIV</code>) is included in a preselected audit class and if the operation involves the use of privilege, then an audit record is always generated. With the previous exception, the read operations for which audit records are not generated when the public flag is set are: <code>access(2)</code>, <code>fgetcmwlabel(2)</code>, <code>fgetslldname(2)</code>, <code>fstatvfs(2)</code>, <code>getcmwfsrange(2)</code>, <code>getcmwlabel(2)</code>, <code>getfpriv(2)</code>, <code>getmldadorn(2)</code>, <code>getslldname(2)</code>, <code>lgetcmwlabel(2)</code>, <code>lstat(2)</code>, <code>open(2)</code>—read only, <code>pathconf(2)</code>, <code>preadl(2)</code>, <code>readl(2)</code>, <code>readlink(2)</code>, <code>stat(2)</code>,</p>

	statvfs(2), mldlstat(3), and mldstat(3). See <i>Trusted Solaris Administrator's Procedures</i> for more details.
gid	Sets the group ID for all objects in the file system. (Because the GID is an object-level attribute that has precedence over any mount-time attributes, setting this is only useful in the rare case when the type of file system being mounted does not have GIDs on its files or directories. In such cases, it is recommended that a value be explicitly specified for the GID.)
uid	Sets the user ID for all objects in the file system. (Because the UID is an object-level attribute that has precedence over any mount-time attributes, setting this is only useful in the rare case when the type of file system being mounted does not have UIDs on its files or directories. In such cases, it is recommended that a value be explicitly specified for the UID.)
slabel	Sets the sensitivity label for all objects in the file system. Specify the sensitivity label in string (text) or hexadecimal format.
forced	Specify one or more forced privileges for all executable files in the file system. Specify symbolic privilege name(s) in a comma-separated list (such as: forced=file_audit, file_chown;) or use all to indicate all privileges. Using none or omitting the keyword results in no forced privileges being applied. For example, the assignment of forced=; results in the default of none being applied. Any forced privileges must be a subset of the allowed privileges. See priv_desc(4) for names of privileges.
allowed	Specify one or more allowed privilege(s) for all executable files in the file system. Specify symbolic privilege names in a comma-separated list (such as: allowed=file_audit, file_chown;) or use all to indicate all privileges. Using none or omitting the keyword results in no allowed privileges being applied. See priv_desc(4) for names of privileges. Any allowed privilege(s) must be a superset of the forced privileges.
low_range	Specify the lower bound of the file system label range as a sensitivity label in string (text) or hexadecimal format.
hi_range	Specify the upper bound of the file system label range as a sensitivity label in string (text) or hexadecimal format.

mld_prefix Set a prefix to be used in the adorned names of multilevel directories. (See *multilevel directories* in the *DEFINITIONS* in *Intro(2)* for more about the MLD prefix.) Specify the value in text format (such as: `.MLD.` or `.hidden.`). On unlabeled (fixed attribute) file systems, the prefix generally has no useful effect—with the exception that an `mld_prefix` should be supplied if a variable filesystem is being mounted on the unlabeled filesystem and the root of the variable filesystem is an MLD.

A comment line or entry is terminated by an unescaped newline character. Lines ending with a backslash (\) continue the current entry to the next line. Leading and trailing white space characters (blank, tab) surrounding a keyword or an attribute value are ignored. When a keyword value is quoted, spaces can be included within the value. Comments are indicated by a pound sign (#) at the beginning of a line and cause the rest of the line to be ignored.

When a keyword appears without an attribute value or when a keyword is missing, a default value is assigned to that attribute. The default values for fixed attribute file systems are:

<code>acc_acl</code>	None
<code>mode</code>	The mode should always be explicitly set for file systems that do not support access modes, such as MS-DOS (<code>pcfs</code> type) file systems.
<code>attr_flag</code>	None
<code>gid</code>	The GID should always be explicitly set for file systems that do not support group IDs, such as MS-DOS (<code>pcfs</code> type) file systems.
<code>uid</code>	The UID should always be explicitly set for file systems that do not support user IDs, such as MS-DOS (<code>pcfs</code> type) file systems.
<code>slabel</code>	The default sensitivity label of a fixed file system being mounted from a local device (such as a hard disk, floppy, or CD-ROM) is the sensitivity label of the device. For an allocated device, the file system is assigned the sensitivity label at which the device was allocated.
<code>forced</code>	None
<code>allowed</code>	None
<code>low_range</code>	ADMIN_LOW
<code>hi_range</code>	ADMIN_HIGH

mld_prefix None

ATTRIBUTES

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtsr

EXAMPLES

EXAMPLE 1 PUBLIC Filesystem

The following example sets a sensitivity label of `PUBLIC` on a file system (`/workspaces`) being mounted from an unlabeled host running the Solaris operating environment. For this to work, `PUBLIC` must be a valid sensitivity label on the local host, the file system must either be automounted or an entry must exist for the file system in the `vfstab(4)` file. Also, entries for the unlabeled host in the `tnrhdb/tnrhtp` files must assign a template to the unlabeled host that specifies a matching default sensitivity label of `PUBLIC`.

```
/workspaces; \
slabel=PUBLIC;
```

EXAMPLE 2 DOS Filesystem

The following example is for a DOS file system named `/no_attributes`, being mounted from a floppy disk. The file system contains an executable that needs the `file_chown` privilege in order to work. The entry sets a UID and GID of 0, a mode of 02777, a public attribute flag, a forced privilege of `file_chown` and allowed privileges equal to `all`. It explicitly sets the `low_range` for the file system to `ADMIN_LOW` and lowers the `hi_range` from the default of `ADMIN_HIGH` to `ADMIN_LOW`.

```
/no_attributes; \
mode=02777; \
attr_flg=public; \
gid=0; \
uid=0; \
slabel=admin_low; \
forced=file_chown; \
allowed=all; \
low_range=admin_low; \
hi_range=admin_low;
```

SEE ALSO

**Trusted Solaris 7
Reference Manual**

getfattrflag(1), getfsattr(1M), setfsattr(1M), getmldadorn(1),
mount(1M), mount_hsfcs(1M), mount_nfs(1M), mount_nfs(1M),
mount_tmpfs(1M), mount_ufs(1M), newsecfs(1M), priv_desc(4)

*Trusted Solaris Audit Administration and Trusted Solaris Administrator's
Procedures*

**SunOS 5.7 Reference
Manual**

setfacl(1), mount_cachefs(1M), attributes(5)

Index

A

audit — audit control file 17, 21
audit trail file
 — audit.log 23
audit_class — audit class definitions 15
audit_class password file 15
audit_control — control information for system
 audit daemon 17
audit_data — current information on audit
 daemon 21
audit_event — audit event definition and class
 mapping file 22
audit_event file 22
audit.log — audit trail file 23
audit_user — per-user auditing data file 31
audit_user password file 31
auth_desc — descriptions of defined
 authorizations 32
auth_name — authorization description
 database 41

C

config.privs — window privileges that override
 system checks 43

D

device_allocate
 device access control file 44
device_deallocate
 device access control file 46
device_maps

 device access control file 48
device_policy — device policy file 50
devices
 access control file — device_allocate 44,
 46, 48

F

file formats
 — intro 11
file system
 defaults — vfstab 135
 mounted — mnttab 67
files used by programs
 /etc/security/device_allocate —
 device_allocate file 45
 /etc/security/device_maps —
 device_maps file 49

I

inetd.conf — Internet server database 54
inittab — script for init 57
Internet servers database — servers 54

L

label_encodings -label encodings file 60

M

mnttab — mounted file system table 67
modified configuration files 11
mounted file system table — mnttab 67

N

name servers
 configuration file — resolv.conf 90
name service switch
 configuration file — nsswitch.conf 68
NFS
 remote mounted file systems — rmtab 94
nsswitch.conf — configuration file for the name
 service switch 68

P

package security attribute description file
 — tsolinfo 125
priv_desc — descriptions of defined
 privileges 75
priv_name — privilege description
 database 88

R

remote mounted file systems
 — rmtab 94
resolv.conf — configuration file for name server
 routines 90
rmtab — remote mounted file system table 94
routing — static, using tsolgateways 121

S

security policy 11

sel_config — selection rules for copy, cut, paste,
 drag and drop operations 95
shared resources, local
 — sharetab 97
sharetab — shared file system table 97

T

tndlog — log of tnd debug information 98
tnidb — trusted network interface-control
 database 99
tnrhdb — trusted network remote-host
 database 103
tnrhtp — trusted network remote-host
 templates 105
tsolgateways — static routing configuration
 file 121
tsolinfo — listing of software package
 contents 125
tsolprof — Trusted Solaris User Profiles
 Database 128
tsoluser — Trusted Solaris User Security
 Attributes Database 132

V

vfstab — defaults for each file system 135
vfstab_adjunct — file mountable security
 information 137