Trusted IRIX®/CMW Security Features User's Guide

Document Number 007-3300-002

CONTRIBUTORS

Written by Jeffrey B. Zurschmeide
Revised by Karen Johnson
Illustrated by Dany Galgani
Production by Kirsten Pekarek
Engineering contributions by Michael Kaye and Eric Lund
St. Peter's Basilica image courtesy of ENEL SpA and InfoByte SpA. Disk Thrower image courtesy of Xavier Berenguer, Animatica.

© 1996, 1998 Silicon Graphics, Inc.— All Rights Reserved
The contents of this document may not be copied or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

RESTRICTED RIGHTS LEGEND
Use, duplication, or disclosure of the technical data contained in this document by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013 and/or in similar or successor clauses in the FAR, or in the DOD or NASA FAR Supplement. Unpublished rights reserved under the Copyright Laws of the United States. Contractor/manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA 94043-1389.

Silicon Graphics, IRIS, IRIX, and the Silicon Graphics logo are registered trademarks and Trusted IRIX/CMW and IRIS InSight are trademarks of Silicon Graphics, Inc. Sun and RPC are registered trademarks and NFS is a trademark of Sun Microsystems, Inc. The X Window System is a trademark of Massachusetts Institute of Technology. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

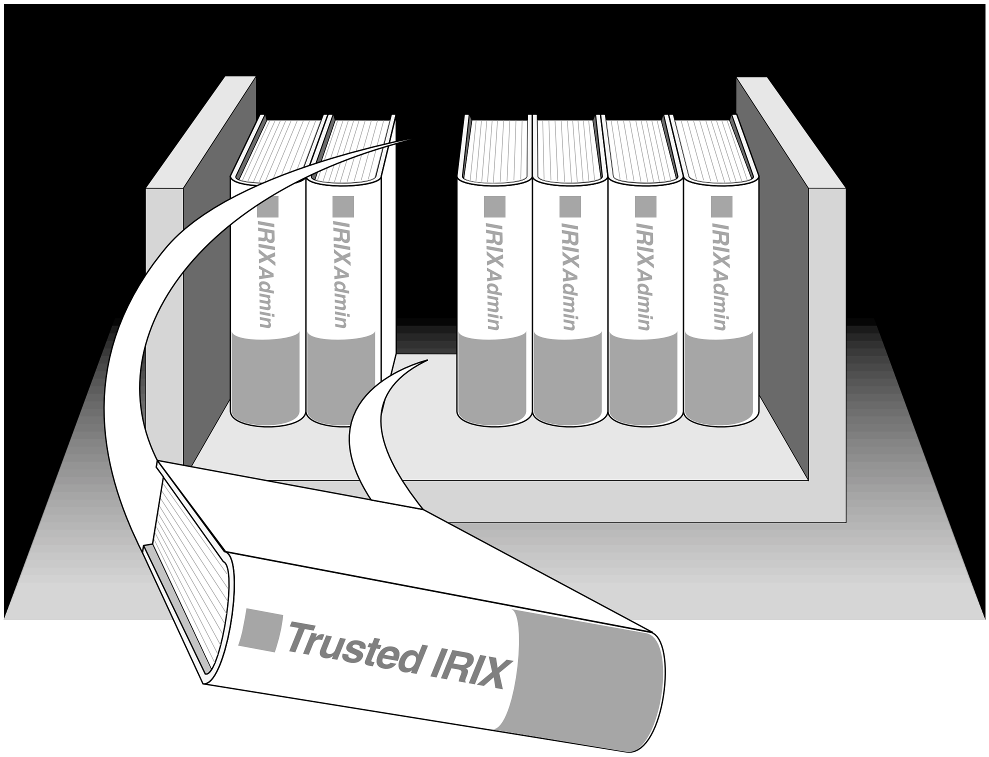
Trusted IRIX®/CMW Security Features User's Guide
Document Number 007-3300-002

Contents

List of Figures

List of Tables

About This Guide



"About This Guide" includes brief descriptions of the contents of this guide and an explanation of the typographical conventions used, and refers you to additional sources of information you might find helpful.

This guide explains how to use the Trusted IRIX/CMW (Compartmented Mode Workstation) operating system with Silicon Graphics workstations and servers. It provides descriptions of those user tasks that are specific to this version of the operating system.

If you have a graphics workstation, you should be familiar with the user documentation of the standard IRIX operating system, on which this product is based. See the *SGI_EndUser* bookshelf in your IRIS InSight online documentation system.

Who Should Read This Guide

You should read this guide if you have never used a secure system before or if you are using Trusted IRIX/CMW for the first time.

Accompanying Documentation

To administer and use the Trusted IRIX/CMW operating system, you must have the set of standard IRIX documentation in addition to the Trusted IRIX/CMW release. In addition to this manual the following document is included:

Trusted IRIX/CMW Security Administration Guide

This manual describes how to administer your Trusted IRIX/CMW site.

Release Notes

This document describes how to install the release and any known problems with the implementation.

What This Guide Contains

This guide contains the following chapters:

Chapter 1, "Introduction to Trusted IRIX/CMW"
Provides an overview of Trusted IRIX/CMW.

Chapter 2, "Getting Acquainted With Trusted IRIX/CMW"
Provides a comprehensive overview of the responsibilities of the user and the operating system features to be used.

Chapter 3, "Understanding Access Control"
Provides information on the mandatory and discretionary access control features of Trusted IRIX/CMW.

Chapter 4, "Understanding System Access"
Describes the tasks and procedures necessary to successfully log in and keep passwords current.

Chapter 5, "Importing and Exporting Data"
Provides information on the security requirements and features relating to media and data import and export generally.

Chapter 6, "Understanding Auditing"
Describes the auditing features and the user's responsibilities with respect to an audited environment.

Chapter 7, "Programming in a Trusted Environment"
Provides information on programming practices in a trusted environment.

Appendix A, "Glossary Of Computer Security Terms"
Provides a glossary of computer security terms and concepts used in these guides and elsewhere.

Conventions Used in This Guide

These type conventions and symbols are used in this guide:

Bold	C++ class names, C++ member functions, C++ data members, function names, language keywords and data types, literal command-line arguments (options/flags), nonalphabetic data types, operators, and subroutines.
Helvetica Bold	Hardware labels
Italics	Backus-Naur Form entries, command monitor commands, executable names, filenames, glossary entries (online, these show up as underlined), IRIX commands, manual/book titles, new terms, onscreen button names, program variables, tools, utilities, variable command-line arguments, variable coordinates, and variables to be supplied by the user in examples, code, and syntax statements
`Fixed-width type`	Error messages, prompts, and onscreen text
`Bold fixed-width type`	User input, including keyboard keys (printing and nonprinting); literals supplied by the user in examples, code, and syntax statements
(ALL CAPS	Environment variables, operator names, directives, defined constants, macros in C programs
“”	(Double quotation marks) Onscreen menu items and references in text to document section titles
()	(Parentheses) Following function names—surround function arguments or are empty if the function has no arguments; following IRIX commands—surround reference page (man page) section numbers
[]	(Brackets) Surrounding optional syntax statement arguments
`#`	IRIX shell prompt for the superuser (*root*)
`%`	IRIX shell prompt for users other than superuser
>>	Command Monitor prompt
>	Cascading menu options: File > Delete

This guide uses the standard UNIX convention for referring to entries in IRIX documentation. The entry name is followed by the section number in parentheses. For example, rep(1C) refers to the *rcp* online reference page.

How to Use This Guide

The *Trusted IRIX/CMW Security Features User's Guide* is written for end users of Trusted IRIX/CMW systems. Frequently, people who would consider themselves end users find themselves performing advanced administrative tasks. For those individuals, the *Trusted IRIX/CMW Security Administration Guide* has been prepared to help both the new and experienced administrator successfully perform all operations necessary to configure and maintain CMW security on Trusted IRIX/CMW systems.

Audience for This Guide

This guide is intended for end users who have never used a secure system before or for those using Trusted IRIX/CMW for the first time.

Additional Resources

For easy reference, this section contains a list of the guides and resources provided with your system and the specific focus and scope of each. You can see the guides by invoking the IRIS InSight library on your desktop or through the system toolchest, or through the *iiv* command from a shell window.

Silicon Graphics End User Documentation

Your IRIS InSight documentation library contains a bookshelf titled *SGI_EndUser*. This bookshelf contains the end user documentation for your system. Some of these books include:

- *IRIS Essentials* or *Desktop User's Guide*
- *IRIS Glossary of Terms*
- *IRIS Utilities Guide*
- *Personal System Administration Guide*
- *Media Control Panels User's Guide*

These books have been written for standard IRIX. Where they differ from information in this book and in the *Trusted IRIX/CMW Security Administration Guide*, the Trusted IRIX/CMW books should be considered authoritative.

IRIX Admin Manual Set

The *IRIX Admin* suite is intended for administrators: those who are responsible for servers, multiple systems, and file structures outside the user's home directory and immediate working directories. If you find yourself maintaining systems for others or if you require more information about IRIX than is in the end-user manuals, these guides are for you. The *IRIX Admin* guides are available through the IRIS InSight online viewing system. The set comprises these volumes:

- *IRIX Admin: Software Installation and Licensing*—Explains how to install and license software that runs under IRIX, the Silicon Graphics implementation of the UNIX operating system. Contains instructions for performing miniroot and live installations using Inst, the command line interface to the IRIX installation utility. Identifies the licensing products that control access to restricted applications running under IRIX and refers readers to licensing product documentation.
- *IRIX Admin: System Configuration and Operation*—Lists good general system administration practices and describes system administration tasks, including configuring the operating system; managing user accounts, user processes, and disk resources; interacting with the system while in the PROM monitor; and tuning system performance.
- *IRIX Admin: Disks and Filesystems*—Describes how to add, maintain, and use disks and filesystems. Discusses how they work, their organization, and how to optimize their performance.
- *IRIX Admin: Networking and Mail*—Describes how to plan, set up, use, and maintain the networking and mail systems, including discussions of sendmail, UUCP, SLIP, and PPP.
- *IRIX Admin: Backup, Security, and Accounting*—Describes how to back up and restore files, how to protect your system's and network's security, and how to track system usage on a per-user basis.
- *IRIX Admin: Peripheral Devices*—Describes how to set up and maintain the software for peripheral devices such as terminals, modems, printers, and CD-ROM and tape drives. Also includes specifications for the associated cables for these devices.
- *IRIX Admin: Selected Reference Pages*—Provides concise reference page (manual page) information on the use of commands that may be needed while the system is down. Generally, each reference page covers one command, although some reference pages cover several closely related commands. Reference pages are available online through the man(1) command.

Reference Pages

The IRIX reference pages (often called “man” or “manual” pages) provide concise reference information on the use of IRIX commands, subroutines, and other elements that make up the IRIX operating system. This collection of entries is one of the most important references for an administrator. Generally, each reference page covers one command, although some reference pages cover several closely related commands.

The IRIX reference pages are available online through the *man* command. To view a reference page, use the *man* command at the shell prompt. For example, to see the reference page for *diff*, enter

```
man diff
```

It is a good practice to print those reference pages you consistently use for reference and those you are likely to need before major administrative operations and keep them in a notebook of some kind.

Each command, system file, or other system object is described on a separate page. The reference pages are divided into seven sections, as shown in Table i. When referring to reference pages, this document follows a standard UNIX convention: the name of the command is followed by its section number in parentheses. For example, *cc*(1) refers to the *cc* reference page in Section 1.

Table i shows the reference page sections and the types of reference pages that they contain.

Table i Outline of Reference Page Organization

Type of Reference Page	Section Number
General Commands	(1)
System Calls and Error Numbers	(2)
Library Subroutines	(3)
File Formats	(4)
Miscellaneous	(5)
Demos and Games	(6)
Special Files	(7)

Release Notes

Release notes provide specific information about the current release and exceptions to the administration guides. Release notes are available online through the *relnotes* command. Each optional product or application has its own set of release notes. The *grelnotes* command provides a graphical interface to the release notes of all products installed on your system.

IRIX Help System

Your IRIX system comes with an online help system that provides help cards for commonly asked questions about basic system setup and usage. The command to initiate a help session is *desktophelp*.

Silicon Graphics World Wide Web Site

The Silicon Graphics World Wide Web (WWW) presence has been established to provide current information of interest to Silicon Graphics customers. The following URL addresses are accessible to most commercially available Web browsers on the Internet:

http://www.sgi.com
: The Silicon Graphics general Web server

http://www.sgi.com/MIPS
: The Silicon Graphics MIPS division server

http://www.ids.sgi.com
: The InterActive Digital Solutions server

http://www.aw.sgi.com
: The Alias/Wavefront server

http://techpubs.sgi.com/library
: The Silicon Graphics Technical Publications Library

From these sites you can find all the Silicon Graphics Web-published information, including the Technical Publications Library.

Chapter 1

Introduction to Trusted IRIX/CMW

This user's guide has been designed to introduce you to working with secure systems. In particular, it introduces you to the Silicon Graphics Trusted IRIX/CMW (Compartmented Mode Workstation) system and provides information on how to maintain system integrity by using security features. It also describes the various modifications and additions made to standard IRIX that make this system secure.

This chapter introduces you to the basic concepts, terms, and features of a trusted system. The following sections are included:

- "Trusted IRIX/CMW Product Overview" on page 20T
- "Trusted IRIX/CMW Security Features" on page 24
- "TSIX Session Manager" on page 29
- "Data Import/Export Restrictions" on page 30

Trusted IRIX/CMW Product Overview

This chapter introduces you to the basic concepts, terms, and security procedures and mechanisms of a trusted system.

What Is a Trusted System

Operating systems that attempt to provide a secure environment for the development and storage of sensitive information are known as *trusted* systems. In an abstract sense, no system is ever perfectly secure from harm, so we use the term *trusted* rather than *secure*. A trusted system can be thought of as any system that fits the following criteria:

- The system allows all users to do their ordinary and necessary work without difficulty.
- The system enforces the security policy deemed by the management to be appropriate to the site.

The first criterion is the most important. If users are unable to do their ordinary and necessary work, they either will circumvent the security measures or they will not use the system at all. In either case, the trusted system is rendered useless. Many users are concerned that they will not be able to do their work in a trusted environment. A good site administration plan structures a trusted system so that the user is relatively unaffected by its functioning. Ideally, users should be able to perform all their tasks and only see the trusted features of the operating system when necessary.

The second criterion requires that the system have adequate security features to enforce the site security policy set forth by the management. Trusted IRIX/CMW offers a variety of security measures that are sufficient to satisfy most sites. These measures are as follows:

Access Control Lists
An Access Control List allows the owner of a file or directory to make a specific list of users and user groups and the specific permissions each one is allowed to the file or directory. ACLs are a standard feature of IRIX.

Auditing
The audit subsystem allows the system administrator to keep a precise log of all system activity. Auditing is a standard feature of IRIX.

Capabiliy A capability is a discreet unit of privilege that can be assigned to a process and allows the process to override a set of related system restrictions.

Capability-based Privilege Mechanism

This is the mechanism through which a privilege is determined based on the set of effective capabilities in a process. Also, it is the mechanism through which capabilities are assigned to a process or an executable file, and through which a process manages its capabilities.

Discretionary Access Control

This is the standard IRIX system of file and directory permissions.

Identification and Authenthication

Trusted IRIX/CMW has improved user identification and authentication facilities that ensure the integrity of system passwords and help to ensure that only authorized users are granted access to the system.

Mandatory Access Control

This facility allows the system administrator to assign security classification labels to files and directories and security clearance labels to users. This is in addition to the Access Control Lists, Capabilities, and Discretionary Access Controls available on the system.

Mandatory Integrity

This is a part of the Mandatory Access Control system that covers an integrity requirement. It allows the system administrator to limit the ability of highly trusted users to access files and programs that are not absolutely secure and trusted.

Mandatory Sensitivity

This is a part of Mandatory Access Control that allows the system administrator to restrict access to files, directories, and programs according to security clearance requirements.

Privileges Privilege is the ability to override system restrictions. This ability is based on an authority that is specific to the privilege mechanism or mechanisms in use by a given site.

Superuser-based Privilege Mechanism

The mechanism through which the IRIX system associates privilege with the root user identity.

Why Use a Trusted System

The Trusted IRIX/CMW system is designed to address the three fundamental issues of computer security: policy, accountability, and assurance. By fully addressing these areas, the system becomes a trustworthy base for secure development and business. Because the nature of a trusted system is already constrained, little must be trusted beyond the system itself. When you run your application programs on the system, you have a reasonable certainty that your applications will be free from corruption and safe from intruders. Trusted IRIX/CMW meets all security requirements for B1 assurance and CMW systems set forth by the National Computer Security Center (NCSC), and all feature requirements through the TCSEC B3 level.

CMW stands for Compartmented Mode Workstation, which means that your individual windows and processes running simultaneously need not all be at the same MAC label. This "compartmentalization" of windows and processes adds greatly to the usability of the system. In all other ways, the system conforms to standard TCSEC B3 feature set, but with assurance of security at the B1 level.

The most important security aspect of the system is a clear definition of the site security policy with respect to all the trusted system features listed above. To accomplish this, all system objects have been examined and altered to close potential security holes and determine a basic clearance level. This examination and revision process ensures the integrity and security of the distributed system.

Another highly important security aspect is assurance. A secure system design must be inspected and approved by a competent agency. Trusted IRIX/CMW from Silicon Graphics is under evaluation for the B1 security rating from the NCSC and Trusted IRIX/B version 4.0.5/epl has been successfully evaluated at the B1 level. IRIX is under evaluation for the C2 security rating.

Why Use Trusted IRIX/CMW

Trusted IRIX/CMW is a significant improvement over conventional trusted operating systems derived from the standard UNIX kernel. While secure operating systems necessarily compartmentalize user interactions, the system need not be hostile to experienced or novice users.

Trusted IRIX/CMW is fully integrated with standard IRIX. IRIX is the Silicon Graphics implementation of the UNIX System V Operating System. Trusted IRIX/CMW is an add-on, developed to conform to the functional requirements set forth in the U.S. National Computer Security Center (NCSC) Orange Book for an A1-level trusted operating system. The Orange Book is a common name for the 5200.28-STD Department of Defense Trusted Computer Systems Evaluation Criteria. Trusted IRIX/CMW will be evaluated at the assurance level as a B1 system.

Ease of Use

As a modified version of an existing operating system, many of the underlying features of Trusted IRIX/CMW have withstood the test of time. Designing a system that promoted "ease of use" was a paramount consideration in the creation of IRIX. Silicon Graphics has a firm commitment to "visual computing," evidenced in the graphical tools provided to you in the IRIX environment.

Greater User-Friendliness

Part of our commitment to ease of use is our commitment to "user-friendliness." A consistent and logical framework underlies the design of Silicon Graphics visual desktop tools. This design permits even the novice user to move about the operating system with some confidence. The desktop provides a visual representation of the filesystem and allows you to navigate using the mouse alone.

Better Support

Silicon Graphics consistently ranks at the top or near the top in customer satisfaction polls. Customer support, in a timely manner, has and will continue to be a corporate goal.

You may contact Silicon Graphics customer support at: 1-800-800-4SGI.

Trusted IRIX/CMW Security Features

The distinguishing difference between trusted systems and nontrusted systems is the security-enhanced feature set. For CMW-level systems, this feature set includes three main components. These components are improved identification and authentication of users, auditing, object reuse, and access control (MAC and DAC).

As well as the required feature set, Silicon Graphics has implemented the X Window System and networking services for the trusted environment. Each component feature is described in detail in this section.

Every trusted system has a Trusted Computing Base (TCB). The TCB is the system hardware, the operating system program itself, and the commands, utilities, tools, and system files that are known to be secure. This set of hardware, files, and programs is the "trusted" part of a trusted system.

Within the TCB, there are *subjects* and *objects*. A subject is any active force on the system, such as a user's shell process, or the audit daemon, or the operating system itself. An object is any passive resource on the system, such as a text file, a page of memory, or a piece of system hardware.

Trusted IRIX/CMW is fully configurable to your site's needs. You are free to select your own security clearances, your own capabilities and access control lists, and your own system of password protection.

Identification and Authentication

The Identification and Authentication (I&A) mechanism controls user access to the system. In common terms, the I&A mechanism is the login procedure. This subsystem is always active if the system is running, and it is impossible to have any contact with the system without first logging in through the I&A system.

The improved I&A facilities of Trusted IRIX/CMW allow the administrator to be certain that the people on the system are authorized users and that private password integrity is maintained to the highest possible levels.

Passwords Under Trusted IRIX/CMW

Under Trusted IRIX/CMW, encrypted passwords are stored separately from other user identification information. This separate location is hidden from normal user access, so the process of a systematic "dictionary encryption" hunt for a password is precluded. User clearance information is also stored in a hidden or shadow file. Under Trusted IRIX/CMW, the */etc/passwd* file does not contain the encrypted password; only the shadow password file contains that information.

In response to extensions to the CMW requirements, passwords can be generated automatically for the users under Trusted IRIX/CMW. The system administrator can configure the system to require this feature for every password change, or it can be an option for the user. The complexity, length, and character combinations required of passwords can also be configured. For example, it is possible to require users to mix control characters into their passwords. It is also possible to check and reject passwords that can be found in a dictionary, proper names, place names, and technical words associated with computers or the current project. System administrators can also require passwords to be changed on a regular basis.

Multilevel Login

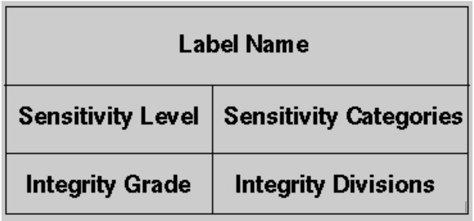
Individual users may have a range of security levels available that have been predetermined by the administrator. The user is not always required to log in at the highest assigned level, thus allowing the flexibility to log in at a level appropriate for a given task. After a successful login has been established, the user may change the clearance of his or her process during the course of the login session. When this happens, all open file descriptors are closed and all objects cleared to prevent declassification or violation of the security policy. All changes of clearance are audited.

Mandatory Access Control

Mandatory Access Control (MAC) allows the administrator to set up policies and accounts that will allow each user to have full access to the files and resources he or she needs, but not to other information and resources not immediately necessary to perform assigned tasks. The access control is called "mandatory" because the system does not allow the owner of the files to change the security classification of system objects. Also, under MAC, access permission cannot be passed from one user to another, as under traditional UNIX systems, which use only Discretionary Access Control. Trusted IRIX/CMW includes both Mandatory and Discretionary Access Control, which work together to precisely control system access.

Under Trusted IRIX/CMW, Mandatory Access Control is divided into two interrelated subsystems: Mandatory Sensitivity and Mandatory Integrity. The access-control enhancements to Trusted IRIX/CMW allow the administrator to set up levels of clearance and related categories of files and other resources, and to assign each user a clearance (or range of clearances). Through this system of access controls, the administrator can custom tailor a user's environment so that the particular user has access only to those files and resources he or she needs to complete required tasks. If there is a breach into that user's account, the unauthorized user has access to very little of the site's protected information.

Each label used for access control has two parts: the sensitivity label and the integrity label. Figure 1-1 shows the components of a label.



Label Name	
Sensitivity Level	Sensitivity Categories
Integrity Grade	Integrity Divisions

Figure 1-1 Basic Trusted IRIX/CMW Security Label Structure

Sensitivity Label Components

Sensitivity labels define the "secretness" or "classification" of files and resources and the clearance level of users. A sensitivity label is composed of a sensitivity *level* and possibly some number of sensitivity *categories*.

There are 256 hierarchical sensitivity levels available for the administrator to create security classifications. In a commercial environment, this label attribute could be used to classify, for example, levels of a management hierarchy. Each file or program has one hierarchical sensitivity level. A user may be allowed to use several different levels, but only one level may be used at any given time.

Over 65,000 sensitivity categories are available for files and programs. For example, categories could include information sorted by subject matter such as geography, demography, astronomy, and others. Each file or user can be a member of any number of categories or of no categories.

Integrity Label Components

While the sensitivity labels identify whether a user is cleared to view certain information, integrity labels identify whether data is reliable enough for a specific user to see. An integrity label is composed of an integrity *grade* and some number of integrity *divisions.*

There are 256 hierarchical grades to classify the reliability of information. For example, data could be classified as an unreliable rumor or as an absolute, confirmed fact.

There are over 65,000 divisions available to classify information based on its source. The source implies probable integrity of the data. For example, sources of data could be divided into Canadian Government, U.S. Government, CBS News, Hearst Publications, and others. In the commercial environment, data sources could be Trade Shows, Press Releases, Conversational, Dataquest, and the like.

Label Name Aliases

Label names are configurable so that specific sites can control naming conventions to meet their special requirements. For example, the site administrator has control of name length (within limits) and could use non-English names, if desired.

Users should only use labels that have label name aliases associated with them. A user who wishes to use a label without a name should request the system administrator to add one. The non-aliased representation of labels can be both verbose and confusing, leading to possible mishandling by the unwary.

MAC Protected Passwords

Encrypted password files and user clearance data is under mandatory access control and restricted to administrative accounts.

Discretionary Access Control

Trusted IRIX/CMW supports the POSIX P1003.1e Draft16 definition for Access Control Lists (ACLs). This draft standard provides for traditional file permission bits working in concert with the more versatile ACLs. Discretionary Access Control (DAC) permissions are defined by the user who owns the file in question. For example, if a user has a personal file in his or her home directory, that user can set the DAC permissions to allow no other users on the system to view, copy, or edit that file. Default DAC permissions for newly created files are set via the *umask* command.

Thus, to gain access to a file that was created by another user, a user must not only have the proper MAC clearance, but must have set the DAC permissions on the file to allow others to access it. DAC permissions should be set in accordance with site security policies.

Default DAC permissions for newly created files depend on the *umask* and on any default ACL entries found in the containing directory.

Access Control Lists

Access Control Lists (ACLs) allow users to specify on a user-by-user basis who may access their files and directories. The purpose of this feature is to provide a finer level of control than is allowed through traditional discretionary access control.

System Audit Trail

A foundation of Trusted IRIX/CMW is the *system audit trail*. The system audit trail provides a means for the system administrator to oversee each important event taking place on the system. The audit trail is useful for tracking changes in sensitive files and programs and for identifying inappropriate use of the system.

The audit trail is generated by additional code in the operating system kernel that notes specific important events, such as file creation, file changes, file removal, invocation of programs, and the login and logout events.

The audit subsystem allows the administrator to create a dynamic record of the system's activity. This record allows the administrator to hold each user strictly accountable for his or her actions. The audit system is completely configurable at any time by the audit administrator.

Audit information must be carefully gathered and protected so that actions affecting security can be traced to the responsible party. Trusted IRIX/CMW records the occurrences of security-relevant events in an audit log. For each event audited, the system records the date and time of the event, the initiating user, the type of event, the success or failure of the event, and the name and security classification of the files or programs used.

The auditing process is transparent to the user. It is important to recognize that when you work on a trusted system, your actions will be audited. You should not, however, be fearful of the auditing process. Its function is to protect you from others who may try to use your user identity for mischief.

Object Reuse Policy

To preclude accidental disclosure of data, display memory and long-term data storage are subject to an object reuse policy and implementation. For example, all system memory is always automatically cleared before it is allocated to another program. Surrendered disk space is also cleaned before it is reallocated.

TSIX Session Manager

The purpose of trusted networking is to properly label data that is imported or exported from the system, and to appropriately enforce the system security policy on that data.

The TSIX standard was created to allow various trusted operating system vendors to interoperate. Under TSIX networking, labeling occurs at two levels. At the Network Level, IP Security Options (RIPSO or CIPSO) are used to route traffic. At the Session Manager Level, SAMP and SATMP are used to send all the Security Attributes required to enforce security policy between systems on the network.

You should contact your administrator to determine the level of networking support available at your site. Some sites may have a very open networking environment with full connection to Trusted IRIX/CMW machines, while others may not allow any connection between trusted and untrusted systems, or even between trusted systems. Your implementation will be unique, and can be explained to you by your administrator.

Data Import/Export Restrictions

NCSC B-level security standards indicate that label information must be preserved when files are placed on magnetic storage media such as tapes. Trusted IRIX/CMW has modified the *tar* command and *cpio* command to include the *M* keyword, to maintain label information on tape media.

Additionally, CMW standards specify that all paper output must be marked with the label of the information printed. Trusted IRIX/CMW line printer software has been modified to add this feature.

Chapter 2

Getting Acquainted With Trusted IRIX/CMW

This chapter details the specific differences a user encounters the first time he or she uses a Silicon Graphics system running the Trusted IRIX/CMW system. The following sections are included:

- "How to Identify the System" on page 32
- "Getting Help With Installation" on page 33
- "Underlying Facets of Trusted IRIX/CMW" on page 33

How to Identify the System

It is possible for you (or one of your programs) to distinguish which environment you are in by using one of the methods described in this section.

Identifying the System Security Options With a Program

From within a compiled program you can use the system call *sysconf*. Refer to the *sysconf*(3C) reference page for more information on this system call.

Identifying the System From a Shell

In order to determine the operating system, execute the *sysconf* command at a shell prompt.

You will see a great deal of output and, towards the bottom of the list, the relevant information in the following form:

```
SYSNAME                          IRIX
HOSTNAME                         DARIUS
RELEASE                          6.2
VERSION                          03092155
MACHINE                          IP22
HW_SERIAL                        1503356586
HW_PROVIDER                      sgi
ACL                              1
AUDIT                            1
CAP                              1
IP_SECOPTS                       1
MAC                              1
```

Refer to the *sysconf*(1) reference page for more information.

Getting Help With Installation

If you need help installing Trusted IRIX/CMW you may refer to the manuals listed in the Preface. Help with individual tools is frequently available from the main menu of the tool (generally, you have access to a *Help* button). In addition, each tool has an associated menu page that should provide the answers you need.

Further help should be available from your system administrator.

Underlying Facets of Trusted IRIX/CMW

A number of facets of Trusted IRIX/CMW are invisible to the user most of the time. These may occasionally affect your environment, so they are explained in this section.

Access Control Lists

ACLs make it much easier to specify who has access to your files. Some programs will be unaware of ACLs; the implementation does not generally require awareness of ACLs. A file's ACL may be set when it is created, if the containing directory has a default ACL.

Auditing

Auditing is constantly in effect on most Trusted IRIX/CMW systems. This is not a cause for user concern, because the purpose of the audit trail is to discern intentional violations of security by malicious intruders, not to spy on legitimate users.

Labeling

It is possible (though not desirable) to create a higher MAC-labeled file in a lower MAC-labeled directory. This file, of course, would not be visible to you at a lower MAC label. Correspondingly, you would be unable to remove this file and it could consume an arbitrary amount of your disk quota until your administrator removes or downgrades the file.

Mail
: A given piece of mail is readable only if it matches your current MAC label. Mail sent to you at a higher MAC label is unreadable until you log in at that label. Indeed, because of these constraints, you would be unaware that you even had mail addressed to you at this higher MAC label. The side effect is analogous to someone placing a higher MAC-labeled file in one of your directories. This higher MAC label mail can consume an arbitrary amount of your disk resources. Mail sent to you outside your label range is not accessible at any time.

If someone sends you mail outside of your allowed clearances, that mail is not delivered. Mail that you send will not be delivered if your current label is outside the label range of the recipient. For example, if you are logged in at system high privilege, and the recipient has only user privilege, mail sent to that user will be rejected.

Multilevel Directories
: One of the constraints of using a secure system is that access to information is restricted by the user's privileges. Multilevel directories (explained in the section above) are one facet of such a system.

Password Aging
: The strictures on passwords in Trusted IRIX/CMW are not limited to the content of the password. The administrator can set a minimum and a maximum amount of time for the use of a given password. It is quite possible that you would try to log in and be unable to do so because your password had expired if you ignored the warnings to change it.

Password Generation
: Trusted IRIX/CMW comes equipped with a password generation program. When you attempt to change your password, this program presents you with several options for your new password. You must choose one of these passwords or choose not to change your password. This feature can be defeated by your Administrator, in which case you are free to select your own password, subject to certain triviality tests.

Chapter 3

Understanding Access Control

Access control is at the heart of a trusted system. Access control allows the administrators to set up policies and accounts that allow each user to have full access to the files and resources he or she needs, but not to other information and resources not immediately necessary to perform assigned tasks.

Under Trusted IRIX/CMW, There are two forms of access control: these are called Discretionary Access Control (DAC) and Mandatory Access Control (MAC). MAC is further divided into two interrelated subsystems, Mandatory Sensitivity and Mandatory Integrity.

The following topics are included:

- "Discretionary Access Control" on page 36
- "Access Control Lists" on page 40
- "Mandatory Access Control" on page 45
- "Using MAC Labels" on page 50

Discretionary Access Control

Discretionary Access Control (DAC) is the name of the standard UNIX system of access permissions that allow the user to control access to files, directories, and other system resources. The owner of any file or other system object can control access to that object, even by those with equal or dominating clearances, by setting the DAC permissions. Additionally, Access Control Lists (ACLs) can be used to provide a finer granularity of control than is provided by the traditional permission bits.

The significant difference between MAC and DAC is that DAC allows untrusted users to control access to their own files and change that access at will. The only user who can override those access decisions is the superuser (root). DAC fills an otherwise unmet need for system security at the personal level. Every file on the system is subject to both MAC and DAC. You must meet both MAC and DAC requirements to access a file.

Using Discretionary Access Control

Trusted IRIX/CMW divides permissions into three categories and users into three relative groups. The three categories of permissions are *read*, *write*, and *execute*. They are denoted as “r” for read, “w” for write, and “x” for execute in long listings of files. Read permission allows you to look at the contents of a file. Write permission allows you to make changes to or remove a file. Execute permission allows you to run the file as a command from your shell prompt.

To get a long listing, enter the *ls -l* command at your system prompt. Thise command shows you more information about the files in the directory than an ordinary listing. Along with the permission information, the *ls -l* command lists the owners of the files and the size of the files and the date they were last modified. Adding the **-D** command-line option to ls displays the ACL for the file or directory as well.

The three relative groups are the owner of the file, the owner’s group, and every other user. If you get a long listing of a directory, you see that the permissions field looks like this: `-rw-r--r--` Each character is separately significant in the permissions listing. Starting at the left, the first character is a dash. A dash in any place means that no permission is granted and the actions associated with that permission are denied. However, in the leftmost place, the contents of that space describes whether the file is a file, directory, or special device file. If there is a dash in that place, the file in question is an ordinary file. If it is a directory, a `d` appears in that space. If the file is a block special device file, a `b` appears in the space, and if the file is a character special device file, a `c` appears there. For more complete information, consult the *ls*(1) reference page or the */usr/include/sys/stat.h* file.

Directory Permissions

Directories use the same permissions as files, but that their meanings are slightly different. For example, read permission on a directory means that you can use the *ls* command to look at the contents of that directory. Write permission allows you to add, change, or remove files in that directory. (However, even though you may have write permission in that directory, you must also have write permission on the individual files to change or remove them, unless you own the directory.) Finally, execute permission on a directory allows you to use the *cd* command to change directories into that directory.

File Permissions

The first series of three places in the permissions field describes the permissions for the owner of the file. Here is an example of a long listing for a file:

```
-rwx------+ 1 owner grp 6680 Apr 24 16:26 shell.script
```

The file is not a directory, so the first space is blank. The characters *rwx* indicate that the owner of the file, *owner*, has read, write, and execute permission on this file. The second series of three spaces describes permissions for the owner's group. In this case, the group is *grp*. Suppose permissions for this file were slightly different, like this:

```
-rwxr-x---+ 1 owner grp 6680 Apr 24 16:26 shell.script
```

In this case, any member of the group *grp* could read or execute the file, but he or she could not change it or remove it. All members of group *grp* can share a pool of files that are individually owned. Through careful use of group read and write permissions, you can create a set of source files that are owned by one person, but any group member can work on them.

The third series of spaces provides for all other users on the system and is called the public permissions.

The plus sign (+) at the end of the permission string indicates that an ACL is in effect for this file. Use the *ls -D* command to view the ACL for the file. Complete discussion of Access Control Lists is found in the section titled "Access Control Lists."

On a large system with several groups, MAC labels do not provide the complete coverage desired. The individual groups can tailor their working set of files by using file permissions and ACLs to share some files. A file that is set to be readable by any user on the system is called *publicly readable*. Remember that even if DAC makes a file publicly readable, a user must still have appropriate MAC clearance to see the file.

Here is a long listing of the sample *Projects* directory:

```
total 410
drw-------+ 1 owner grp 48879 Mar 29 18:10 critical
-rw-r--r-- 1 owner grp 1063 Mar 29 18:10 meeting.notes
-rw-rw-rw- 1 owner grp 2780 Mar 29 18:10 new.deal
-rwxrwxrwx 1 owner grp 8169 Jun 7 13:41 new.items
-rw-rw-rw- 1 owner grp 4989 Mar 29 18:10 outside.response
-rw------- 1 owner grp 23885 Mar 29 18:10 project1
-rw-r----- 1 owner grp 3378 Jun 7 13:42 saved_mail
-rw-r--r-- 1 owner grp 2570 Mar 29 18:10 schedules
-rwxrwxr-x+ 1 owner grp 6680 Apr 24 16:26 shell.script
```

The files have varying permissions. Some can be read and written to only by the owner, some can be read only by members of the owner's group, and some can be read, changed, or removed by anybody. The shell script can be executed publicly, subject to its ACL, and the *critical* directory is also subject to an ACL.

Changing Permissions

You change the permissions on a file by means of the *chmod* command. You can use *chmod* only to change files that you own. Generally, you use this command to protect files you want to keep secret or private, to protect private directories, and to grant permissions to files that need to be used by others. The command to restrict access to a file or directory to yourself only is:

```
chmod 600 filename
chmod 700 directoryname
```

Other permissions may be added by using the *chmod* command with the letter associated with the permission. For example, the command to add general write permission to a file is

```
chmod +w filename
```

For more examples, see the *chmod*(1) reference page.

To set or change an ACL, use the *chacl* command:

```
chacl  acl_entry  [, acl_entry ] ...
```

For more information on *chacl* and the acl entry syntax, see the chacl(1) reference page and the section of this chapter titled “Text Form Representation of ACLs.”

Setting Permissions with umask

You can assign default permissions to your files by using the *umask* command. Place this command in your *.cshrc*, *.profile*, or *.login* file. The *umask*(1) reference page is also available for more information. By changing the setting of your *umask*, you can alter the default permissions on your files and directories to any available DAC permission.

A drawback to the *umask* command is that it makes every file you create receive the same permissions. For most purposes, you want the files you create to be accessible by the members of your group. For example, if an individual is suddenly called away and another person must take over that person’s portion of a project, the source files must be accessible by the new user. However, the personal files you keep in your home directory sometimes need to be private, and if you set your *umask* to allow group read and write privileges, any member of the group can access your personal files. Mechanisms are available to prevent this access. For example, you can create a directory of private files and alter the permissions on that directory with the *chmod* command to restrict all but your own access. Then it would not matter that the files were readable, because no other user would be allowed into the directory.

You can also use the *find* command to change all the files in your home directory to your chosen permission automatically at your convenience. You can set up your account so that this action happens every time you log out.

The *umask* command is an important part of DAC. It allows us to maintain security and still allow convenient access to your files. To set your account up to allow group read and write privileges and no other privileges, place this line in your *.cshrc* or *.profile* file:

```
umask 007
```

This will make every file you create have the following permissions:

```
-rw-rw----
```

With your *umask* set to 007, directories that you create have the following permissions:

```
drwxrwx---
```

In plainer terms, you will have full use of the file or directory, and your group will have full use. No other user, except the superuser (root), will have access to your files.

Access Control Lists

Access Control Lists (ACLs) are a part of the DAC features of your Trusted IRIX/CMW system. An ACL works in the same way as standard file permissions, but it allows you to have a finer level of control over whom may access a file or directory than standard permissions allow. ACLs allow you to specify file permissions on a user by user basis.

Every system file or directory has an ACL that governs its discretionary access. This ACL is referred to as the access ACL for the file or directory. In addition, a directory may have an associated ACL that governs the initial access for files and subirectories created within that directory. This ACL is referred to as a default ACL. A user who wishes to gain access to the files in a directory must be on both ACLs and must be allowed by MAC and Trusted IRIX standard file permissions to successfully gain access. If you have not created an access ACL for a file, the default ACL serves both ACL functions. Note that the ACL on a file or directory also acts as an upper limit to the file permissions that can be set automatically with *umask*.

Hereafter in this section, directories are treated as files, and where the term file is used, consider it to also apply to directories.

An ACL is stored in the same way that standard file permissions are stored; as an attribute of the file or directory. To view the ACL of a file, use the **-D** option to *ls* as shown here:

```
ls -D /usr/people/ernie/testfile
```

This produces output similar to this:

```
testfile [u::rwx,g::rw-,o::---,u:332:r--,u:ernie:rw--,m::rw-
```

The above example shows full permissions for the owner with the first entry on the line, sets read permission for user ID 332 with the second entry, and sets read and write permission for the user account ernie. The format of an ACL entry is discussed in the section titled “Text Form Representation of ACLs.”

To set or change an ACL, use the *chacl* command:

```
chacl  acl_entry[,acl_entry]...
```

An ACL consists of a set of ACL entries. An ACL entry specifies the access permissions on the associated file for an individual user or a group of users. The order of internal storage of entries within an ACL does not affect the order of evaluation. In order to read an ACL from an object, a process must have read access to the file. In order to create or change an ACL , the process must own the file.

Setting Directory Default ACLs With chacl

To set a default ACL for the current directory and all its files and subdirectories, use this command:

`chacl -d` *acl_entry*`[,`*acl_entry*`]...`

For information on the format of an ACL entry, see the section titled “Text Form Representation of ACLs.”

Text Form Representation of ACLs

This section defines the long and short text forms of ACLs. The long text form is defined first in order to give a complete specification with no exceptions. The short text form is defined afterwards because it is specified relative to the long text form.

Long Text Form for ACLs

The long text form is used for either input or output of ACLs and is defined as follows:

acl_entry`[,`*acl_entry* `]...`

Although it is acceptable to place more than one entry on a physical line in a file, placing only one entry per line makes it easier to read.

Each entry contains one ACL statement with three required colon-separated fields and an optional comment:

entry tag type`:`*entry qualifier*`:`*discretionary access permissions* `#` *comment*

Comments may be included with any entry. If a comment starts at the beginning of a line, then the entire line is interpreted as a comment. The first field must always contain the ACL entry tag type.

One of the following ACL entry tag type keywords must appear in the first field:

user A *user* ACL entry specifies the access granted to either the file owner or to a specified user account.

group A *group* ACL entry specifies the access granted to either the file-owning user group or to a specified user group.

other An *other* ACL entry specifies the access granted to any process that does not match any user, group, or implementation-defined ACL entries.

mask A *mask* ACL entry specifies the maximum access that can be granted by any ACL entry except the user entry for the file owner and the *other* entry.

The second field contains the ACL entry qualifier (referred to in the remainder of this section as simply *qualifier*).

The following qualifiers are defined by default:

uid This qualifier specifies a user account name or a user ID number.

gid This qualifier specifies a user group name or a group ID number.

empty This qualifier specifies that no *uid* or *gid* information is to be applied to the ACL entry. The entry applies to the file owner only. An empty qualifier is represented by an empty string or by white space.

The third field contains the discretionary access permissions that are to apply to the user or group specified in the first field. The following symbolic discretionary access permissions are recognized in ACLs:

- read access
- write access
- execute/search access
- no access

The discretionary access permissions field must contain exactly one each of the following characters in the following order:

1. r
2. w
3. x

This would appear as `rwx`. Any or all of these may be replaced by a dash (-), which is the no-access character.

A user entry with an empty qualifier specifies the access granted to the file owner. A user entry with a *uid* qualifier specifies the access permissions granted to the user name matching the *uid* value. If the *uid* value does not match a user name, then the ACL entry specifies the access permissions granted to the user ID matching the *uid* value.

A group entry with an empty qualifier specifies the access granted to the default user group of the file owner. A group entry with a *gid* qualifier specifies the access permissions granted to the group name matching the *gid* value. If the *gid* value does not match a group name, then the ACL entry specifies the access permissions granted to the group ID matching the *gid* value. The *mask* and other entries contain an empty qualifier. A pound sign (#) starts a comment on an ACL entry. A comment may start at the beginning of a line, or after the required fields and after any custom-defined, colon-separated fields. The end of the line denotes the end of the comment.

If an ACL entry contains permissions that are not also contained in the *mask* entry, then the output text form for that entry must be displayed as described above followed by a pound sign (#), the string `effective:` , and the effective file access permissions for that ACL entry.

White space is permitted (but not required) in the entries as follows:

- at the start of the line
- immediately before and after a colon (:) separator
- immediately before the first pound sign (#) comment character
- at any point after the first pound sign (#) comment character.

Comments have no effect on the discretionary access check of the object with which they are associated.

Here is an example of a correct long text form ACL for a file:

```
user::rwx,user:332:r--,user:ernie:rw-
```

The above example sets full permissions for the owner with the first entry on the line, sets read permission for user ID 332 with the second entry, and sets read andwrite permission for the user account ernie.

Here are some examples with comments:

```
group:10:rw- # User Group 10 has read/write access
other::--- # No one else has any permission
mask::rw- # The maximum permission except for the owner is read/write
```

Short Text Form for ACLs

The short text form is used only for input of ACLs and is defined as follows:

acl_entry[,*acl_entry*]...

Although it is acceptable to place more than one entry on a physical line in a file, placing only one entry per line makes it easier to read.

Each line contains one ACL entry, as defined in “Long Text Form for ACLs” with two exceptions. The ACL entry tag type keyword must appear in the first field in either its full unabbreviated form or its single-etter abbreviated form.

The abbreviation for user is `u`, the abbreviation for group is `g`. The abbreviation for other is `o`, and the abbreviation for mask is `m`.

There are no exceptions for the second field in the short text form for ACLs. The discretionary access permissions must appear in the third field in either absolute symbolic form or relative symbolic form.

The relative symbolic form must be preceded by a plus sign (+) to indicate additional access or a caret (^) to indicate that access is to be removed. The relative symbolic string must be at least one character.

The symbolic string contains at most one each of the following characters in any order:

- r
- w
- x

For example, the short form should look very similar to the following:

```
u: :rwx # The file owner has complete access
u:332:+r # User Acct 332 has read access only
g:10:rw- # User Group 10 has read/write access
u:653:^w # User Acct 653 (who is in group 10) has read access only
o::--- # No one else has any permission
m::rw- # The maximum permission except for the owner is read/write
casey:all=:all+eip
```

```
chcap CAP_AUDIT_WRITE,CAP_AUDIT_CONTROL,CAP_KILL+eip
```

Mandatory Access Control

One of the new features in Trusted IRIX/CMW and in B-level trusted systems that is not available in standard IRIX is Mandatory Access Control (MAC). MAC is essentially different from DAC in that the restrictions placed on file and resource access are not up to the discretion of the individual user, but are mandatory for all users. The system enforces MAC through the security labels of all files, programs, resources, and processes (including user processes) on the system. The concept of label domination and equivalence is used to make MAC decisions. After the sections describing the subdivisions of MAC, there is a section describing the rules of label domination and equivalence.

MAC is divided into two parts, Mandatory Sensitivity and Mandatory Integrity. These two concepts work in concert to provide a trusted environment for the users.

Mandatory Sensitivity

Mandatory Sensitivity (MSEN) is a mechanism for implementing strict controls on access to data. A privileged user can never give information protected by Mandatory Sensitivity to someone who is not allowed to see it. Under DAC, a user can change a file’s permissions so that any user can read, write, or execute the file. This system provides a good level of security in an open system but does not provide the level of security needed by Trusted IRIX/CMW. MSEN works in addition to DAC to provide an extra level of security.

MSEN defines two different kinds of permissions. One kind is for the user and the user's login shell process; the other is for system objects, such as files. The first kind of permission, for users and processes, is called a *clearance*. A clearance permits a user or the user's process to use system objects with corresponding *classifications*. All of the processes that run on behalf of a user must be within the user's clearance.

Each clearance for a user and the processes associated with that user contain a level of clearance, such as *confidential* or *proprietary*. Each user's clearance can also be valid in a number of *categories*. These categories are used to divide files and information logically by relationship. For example, all development files could be in the category *ENGR* and all personnel files could be in the category *HR*. A user with clearance in the *ENGR* category would not necessarily have clearance in the *HR* category, even if the two categories are currently running at the same classification. The number and names of your clearances and categories are configurable at any time.

The combination of clearance and categories forms the MSEN label of a user or a user's process, while the combination of classification and category forms the MSEN label of an object.

An object (a file or system resource) is classified at a level of protection based on the judgment of some person. It is also defined to be in some number of categories. For example, employee salary records could be classified as *top secret* and in the categories *HR*, *management*, and *finance*. Thus, a user who is cleared for *top secret* data in the categories of *HR*, *management*, and *finance* could view the data, but a user cleared only to the level of *secret* could not. A user cleared to *top secret* in another category, such as *ENGR*, also could not view the information. To view information that has categories, you must also be cleared for the same or a strict superset of categories. For example a user cleared to *top secret* in only one category in our example, say *finance*, could not view the employee salary information.

For a person to access a *secret* file about employee records, the user must be cleared for both that level of secrecy and the category of information. Users cleared to levels higher than the level of a given file can also view the file. For example, a user cleared for *top secret* information can read a *secret* file, provided that the user is cleared in the proper category.

Mandatory Integrity

The Mandatory Integrity (MINT) system protects important users from files of questionable integrity. Until a program has been certified to be free of security risks, important users should not be allowed to execute it. Mandatory integrity enforces this restriction.

The MINT mechanism allows read and execute access only to those processes whose integrity labels are dominated by the object (meaning that the file or program has equal or greater integrity than the user process). Additionally, a process may only write to an object with the same integrity. This is to avoid reducing the integrity of a file by a user with lower integrity.

Mandatory Integrity is similar to MSEN in design and implementation, but addresses different issues and threats. While MSEN prevents a user from accessing information that is too sensitive or secret for the user's clearance, MINT prevents a user from accessing information or programs that are of unknown or lower quality or security. For example, a user running at the highest possible clearance who has access to the most secret and important system resources should not be allowed to run every program found on the system. Such a user should be permitted to execute only programs of known good integrity. This step further prevents Trojan Horse attacks on the system.

Consider the following scenario: A malicious intruder gains access to the system but only at the lowest level. This person creates a program to remove or publish certain system files and leaves the program in a public directory, calling it *run.me*. If a high-clearance user finds the file and executes it, serious damage could result. The solution is for the system to attach an integrity label to each file, indicating the known security of the file. A file created by a low-clearance user, such as our intruder, would automatically get a low-integrity label from the system. Any user with higher clearance would not see the low-integrity file when listing the directory contents, and any attempt to run the program would be denied access. Then, the auditor or system administrator would be notified of the denied access through the system audit trail, and the program could be safely removed.

Remember that a user’s integrity requirement does not prohibit accessing files of greater integrity, only those of lower integrity.

MINT divides the objects of the system into *divisions* and assigns each file and resource a *grade*. MINT divisions need not be related to the categories used by MSEN on your system. For example, MINT divisions could be programming tools, general utilities, and administrative utilities. Thus, a user who has a MSEN clearance for ENGR might have a MINT requirement in programming tools, and in general utilities, but not in administrative utilities.

Label Domination and Equivalence

The concept of label domination and equivalence is central to MAC. If a user’s label clearance is higher than a file’s label classification and the integrity grade on the label of the file is good enough for the user’s label, the user’s label is said to dominate the file’s label. If the clearance and classification on both labels are equal, the labels are said to be equal. A user’s label must be at least equal to or must dominate an object’s label in order to access the object.

When you add categories to MAC, you change the order of dominance on your system. In order to dominate, a user’s label must have the same or higher sensitivity and a set of approved categories that are the same as or a superset of the categories of the file’s label, and the integrity requirement for the user must be met by the file. Also, the integrity divisions of the user must be the same or a superset of the integrity divisions of the file.

Table 3-1 lists possible label relationships using the default labels supplied with your system. In the table, the levels of sensitivity are *unclassified*, *proprietary*, and *company sensitive*. The categories are *green*, *gray*, and *gold*. The integrity grades are *good*, *choice*, and *prime*. The integrity divisions are *cake*, *cookie*, and *cracker*. The labels are written in the form of *sensitivity level-categories*, *integrity grade-divisions*.

Table 3-1 Sample Label Relationships

Subject Label	Object Label	Dominates?	Explanation
proprietary/good	unclassified/prime	Yes	Clearance dominated; integrity dominated
proprietary/prime	unclassified/good	No	Integrity of the file not good enough
proprietary,green/ good	unclassified-green/ good	Yes	Clearance dominates; categories equal; integrity equal
proprietary,green/ prime,cake	proprietary-green/ prime,cake, cookie, cracker	Yes	Clearances identical integrity divisions dominate
proprietary/green, prime	company sensitive,green/ prime	No	Object classification higher than user clearance
proprietary,green/, prime	proprietary,green, gray/prime-cake, cookie	No	Categories not equal or dominated
proprietary,green, gray/ prime,cake,cookie	proprietary,green, gray/prime,-cake, cookie	Yes	Categories equal; integrity equal
proprietary,green, gray,gold/choice	proprietary,green, gray/prime	Yes	Categories dominated; integrity dominated

Wildcard Labels

Wildcard labels are special labels for system objects that are always equal to the label of any user process or other system subject that attempts access. For example, many system networking services are implemented through wildcard labels, so that all users can access the service. For example, the */dev/null* device has a wildcard label.

Using MAC Labels

While using Trusted IRIX/CMW, you must change your security label from time to time (if you are cleared for more than one label). You must also change the security label of a file from time to time and you must frequently check the label of a file or resource. There is a group of commands that allow you to perform these activities easily.

Changing Your Security Label

Sometimes you will find it necessary to run a program or other process at a label different from your current login label. For example, the process may require a lower integrity requirement or a higher clearance. The *newlabel* command allows you to run a process at a different label.

To prevent inappropriate transfers or disclosures of information, all open file descriptors associated with your login shell process are closed before the new process is invoked. This assures that information at a higher classification will not be used as any input to the new process, which may be running at a lower clearance. The default new process is your default command shell, as specified in your environment.

Remember that you can execute *newlabel* only with a specified clearance up to the maximum allowed for your login account. For complete information about *newlabel*, consult the *newlabel*(1) reference page.

To execute this command, enter:

```
newlabel label command
```

label is the new security label you want and *command* is the command to be run at the new label. Assuming the label you have chosen is within your label range, the label is changed immediately for the duration of the command. Remember that only root (the superuser) can use *newlabel* to run a shell.

Changing the Label of a File

You are allowed to change the label of any file or program you own, so long as you only upgrade the sensitivity label of the file or downgrade the integrity label. That is, the new label cannot be less sensitive or of higher integrity than the old label. What Trusted IRIX/CMW does when you change the label is to make a copy of the file at the new label, thus allowing the system administrator to undo your change, if necessary. When you make the change, the new label of the file must be equal to the current label of the user attempting the change. Use the *chlabel* command like this:

`chlabel` *label filename*

filename is the name of the file to be changed and *label* is the new label for the file. The *chlabel* command allows you only to change the label to a label within your clearance range. Remember that the label of the directory that contains the file will not be changed, making future deletion or modification of the file impossible without administrator intervention. It is generally better to upgrade whole directories than individual files.

Determining the Label of a File

The **-M** flag to the *ls* command displays the security labels for all files and subdirectories in the directory being listed. Note however, that only those files with labels dominated by your current label will appear in any directory listing, with or without the **-M** flag. If you ever find yourself in a situation where a file seems to have "disappeared", check your label and make certain that the label of the file in question is dominated by your label.

Multilevel Directories

Directories are subject to MAC just as any text file or other resource. Most directories have labels that are identical to any file label. The exceptions are called *multilevel directories* (which are sometimes called *moldy* directories or *mld*).

An mld places the files from each label into multiple hidden subdirectories. Thus, user A at label Q will get a different listing of the contents of the mld from user B at label X. However, neither process will see the subdirectory structure. Each process sees only those files in the mld that have the same label as the process.

The hidden subdirectories in an mld are visible to a user process that has a moldy label. A user may spawn a process with a moldy label using the **-m** option of the *newlabel* command.

Chapter 4

Understanding System Access

This chapter describes the access rules that govern Trusted IRIX/CMW. It includes a step-by-step description of how to log in, a discussion about dealing with the password mechanisms, an explanation of areas where Trusted IRIX/CMW differs from standard IRIX, and short descriptions of some day-to-day tasks that users of Trusted IRIX/CMW will need to perform. For a complete new-user tutorial on all aspects of the IRIX system, refer to your standard IRIX documentation.

The following sections are included:

- "Logging In" on page 54
- "Determining the Security Features of a System" on page 56
- "Using Aliases For Labels" on page 56
- "Passwords Under Trusted IRIX/CMW" on page 57

Logging In

When no one is logged in to a Trusted IRIX/CMW machine, the system displays a login prompt and waits for a user to enter a login name. To log in, you must first have an account created for you on the system. Your system administrator should create this account for you and tell you the login name you are to use. If you are allowed to select your own login name, select a name that is easy to remember, such as your first name and the initial of your last name. When your account is created, a password may also be logged for you at that time by the system adminstrator. If so, you should know the password before you attempt to log in. If a password is not logged for you when your account is created, you will have to select one when you first log in.

When you are certain that the account has been created for you, you are ready to log in. When no one is logged in at the console, a window is displayed for the login dialog. Follow these instructions to log in:

1. The trusted path window is displayed on the screen, as shown in Figure 4-1, and the trusted path should be initialized on.

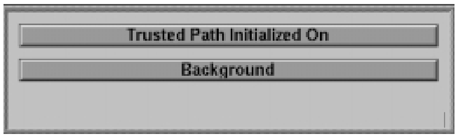
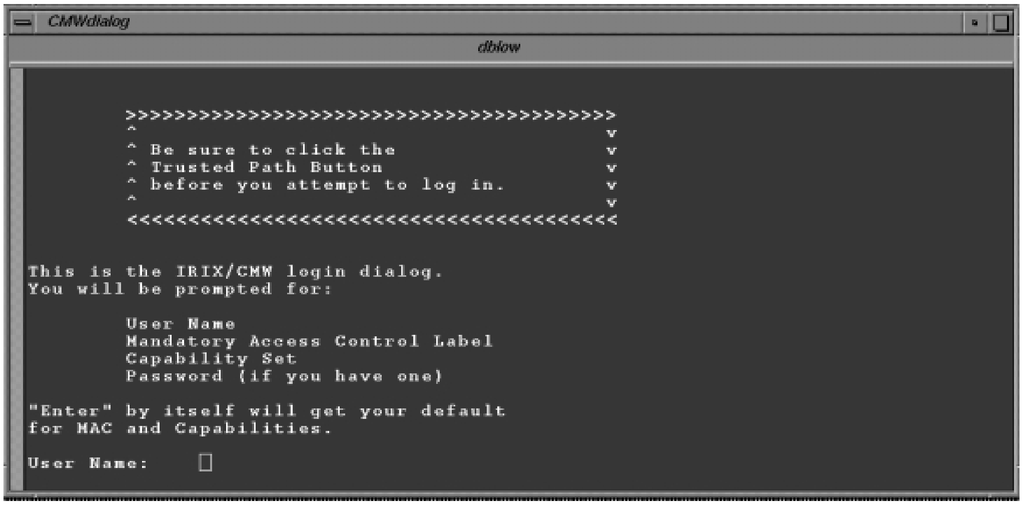


Figure 4-1 Trusted Path Window

If the trusted path is not on, move the mouse cursor to the top button on the trusted path menu and click. If the trusted path window does not indicate that the trusted path is on, call your system administrator. Move the pointer to the CMW Login Dialog window. The trusted path window should state that "You Are On The Trusted Path". Again, if it does not state that you are on the trusted path, call your system administrator.

2. On the CMW Login Dialog window, you see the `User Name:` prompt, as shown in Figure 4-2:



```
        >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        ^                                      v
        ^ Be sure to click the                 v
        ^ Trusted Path Button                  v
        ^ before you attempt to log in.        v
        ^                                      v
        <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<


This is the IRIX/CMW login dialog.
You will be prompted for:

        User Name
        Mandatory Access Control Label
        Capability Set
        Password (if you have one)

"Enter" by itself will get your default
for MAC and Capabilities.

User Name:    
```

Figure 4-2 CMW Login Dialog Window

Enter the desired account name. You must enter an account name; there is no default.

3. You are prompted for a MAC label:

   `MAC Label:`

   If you do not enter a MAC label name (that is, if you simply press the Enter key) you are given your default login label.

4. You are prompted for a capability set:

   `Capabilities:`

   If you do not enter a capability set (that is, if you simply press the Enter key) you are given your default capability set.

5. You are prompted for your password:

   `Password:`

   Your password is not displayed as you type it in. Press the Enter key when you have typed your password.

6. If all responses were valid, you are logged in. If the *login.options* file contains the following line, you are notified of your last login date and time.

   ```
   lastlog = 1
   ```

   This is available so that you can be instantly aware if someone else has logged in to the account since your last login. If you have never logged in before, the system does not display any `lastlog` message.

7. The screen clears and the default windows and icons are displayed. The login process is now complete.

Determining the Security Features of a System

As described before, the *sysconf* command is used to determine the current operating system. A complete description of all *sysconf* command options can be found in the *sysconf*(1) reference page.

Using Aliases For Labels

A label name may be specified for any desired pair of sensitivity and integrity. The format of such an alias is:

aliasname`:[`*msentype*`] [`*level*`[,`*category*`]...]/[`*minttype*`][`*grade*`[,`*division*`]...]`

If you do not supply the msentype field, the type is recorded as TCSEC. If you do not supply the minttype field, the type is recorded as BIBA.

Trusted IRIX/CMW allows the system administrator to create aliases for commonly used labels. For example, we suggest the use of *userlow*, *usermiddle*, and *userhigh* as three labels for three classes of users. Your system administrator should tell you what, if any, label aliases are available at your site. A valid label alias can always be used in place of the specific label name, whether during the login process or when using the system.

Passwords Under Trusted IRIX/CMW

Passwords are the first line of defense of a trusted system. As a user, it is your responsibility to protect the privacy of your password at all times. Follow these rules regarding your password:

- Never give your password to another user or allow another user to "borrow" your account.
- Never keep your password written down anywhere near your machine.
- Always commit your password to memory. If you forget it, the system administrator can change it for you.

Trusted IRIX/CMW contains facilities to generate passwords for users and these facilities are configured to work by default. If your site has changed the configuration to allow you to select your own passwords, follow these rules when choosing your password:

- Never choose a password that could be guessed by someone who knew personal information about you. For example, if someone stole your wallet with the intent of finding out information about you, make certain that your password is not related to something someone might find in your personal information, such as variations on your name or the name of a friend or family member.
- Always use a random mix of printable characters, control characters, punctuation marks, and numerals when selecting a password.
- Each password must have at least six characters. However, only the first eight characters are significant.
- The password must contain at least two alphabet characters and one numeral character.
- The password must not be related to the user's login name. Any reversing or circular shift of the characters in the login name will not be allowed. Capital letters are assumed to be equivalent to their lowercase counterparts.
- The password must have at least three characters different from the previous password. Capital letters are assumed to be equivalent to their lowercase counterparts.

System-Generated Passwords

Trusted IRIX/CMW supports mandatory password generation. The default generator presents the user with five selected passwords, and the user is free to accept or reject any of these. If the user does not accept any of the offered passwords, he or she may press the *Enter* key and the system presents a new set of password choices.

Password Aging

Trusted IRIX/CMW supports password aging. Password aging is defined as being able to set a minimum and maximum lifetime for passwords. Password aging is a very useful feature. By limiting the amount of time a password may be in use, you limit the amount of time a potential intruder has to crack your password. By enforcing a minimum lifetime, you prevent lazy users from simply changing their passwords briefly and then returning to their usual passwords immediately.

If a user does not change his or her password within the specified time period, the account is automatically locked. Any user can place the following line in their *.login* or */.profile* files to notify them when password expiration is imminent:

```
showpwage username
```

By default, *showpwage* only notifies the user if the password is within seven days of expiration. This default can be changed with the *-d* flag. See the *showpwage*(1) reference page for a complete description of this command.

Generally, the only time that an account becomes locked is when the user is away for an extended period of time. But once locked, an account can be unlocked only by the superuser. Then, the system administrator should force the user to choose a new password the next time he or she logs in.

Chapter 5

Importing and Exporting Data

Importing and exporting information is one of the main functions of a computer system. Whenever data enters the computer, it is considered to have been imported from somewhere, whether from the keyboard, the tape drive, or other input device. Anytime the system produces information, such as via a printer or a write action to a tape or floppy disk, an export is considered to have occurred. This chapter describes the restrictions associated with using printers and removeable media under Trusted IRIX/CMW.

Sections in this chapter include:

- “Printing Under Trusted IRIX/CMW” on page 60
- “Using Tape Devices” on page 60

Printing Under Trusted IRIX/CMW

Printing under Trusted IRIX/CMW requires no special resources. Except where noted in this chapter, printing operates exactly as described in your standard IRIX documentation. The *lp* command behaves differently from its IRIX counterpart. You are encouraged to read the lp(1) reference page before using this command.

Trusted IRIX/CMW meets the requirement for B1-level systems for labeled printing. Each page of printed output carries the label of the printing process at the top and bottom of the page. The system intercepts the output of a print request before it is sent to the printer and ensures that appropriate banner pages and individual page labels are produced. Your system administrator will tell you which commands to use to print your files.

Printing Files With Numeric Labels

It is possible that a print job may be submitted with a label that does not appear in the label naming databases. If this occurs, the label printed on the banner pages and at the top and bottom of each page will reflect the numeric values of the component or components that lack assigned ASCII representations. This does not represent a breach of system security. These numeric components accurately reflect the label components they represent; there is simply no meaningful name for that component value. It is important that any print job that has such a label be handled with extreme care because its sensitivity, while displayed accurately, may not be obvious.

If you encounter such a printed label, contact your system administrator.

Using Tape Devices

Under Trusted IRIX/CMW, access to the tape device is administratively controlled. The system administrator must take specific steps to ensure that the tape device is properly configured for your use before you insert the tape in the drive. The procedures required of the administrator are described in the *Trusted IRIX/CMW Security Administration Guide*.

Notify your system administrator that you need to use the tape device and provide the security label of the information you wish to archive and the label your process will have while you use the tape device. The administrator will then have to change the security label of the tape device for you before you can begin. When you are done, the administrator will change the label of the tape drive back to its default. The default label for the tape device is *dbadmin*, which is accessible only by the root account.

Your site may have specific policies regarding the secure handling of tapes, particularly in the area of human-readable “sticky” labels. Your site may require that tapes be handled only by the operator, or you may be allowed to do so yourself.

Once you have made your tape, you must write the security classification and categories, as well as any MINT grades and divisions on it, and handle and store the tape according to your site’s security policies.

Check the local policy with your system administrator before attempting to physically mount a tape.

The basic rules most sites follow for tape handling include:

- Storing the tapes in a locked room, sorted according to security label.
- Limiting access to the tape storage area to people with the highest security clearances.
- Disposing of used tapes in a secure manner, after they have been erased and verified that no information remains readable on the tape. Sometimes tapes are destroyed by burning.

Magnetic Tape Backups With tar

B1 systems are required to provide for labeled magnetic tape archives. Trusted IRIX/CMW meets this requirement by providing the new **M** keyword to the *tar* command. This keyword directs *tar* to maintain the security labels, access control lists, and capability requirements on all files placed on the tape. To recover files from the tape, use *tar* with the **M** keyword. Restoring tapes with files of differing labels requires special capabilities.

Always remember that it is still possible to make unlabeled tapes using *tar* without the **M** keyword. Also, using *tar* to extract labeled files without the **M** keyword will result in the loss of label and other security data.

Chapter 6

Understanding Auditing

This chapter describes the system audit trail for the user. There is no interface to allow users to alter or read the audit trail; it is accessible only to the system administrator or auditor . This chapter explains what is happening within the audit system and how it applies to the ordinary user.

System Audit Trail

The system audit trail (SAT) is a subsystem that allows the site administrator to make a record of all system activity. The ongoing record of system activity shows general trends in system usage, and also violations of the security policy. The site administrators can monitor all system activity through the audit trail. There are many different types of activities that take place on a trusted computer system. There are login attempts, file manipulations, use of devices (such as printers and tape drives), and administrative activity. All of these activities can be logged and reviewed through the system audit trail.

It is vitally important to remember that the system audit trail does not exist to allow users to spy on one another, nor does it exist as a mechanism to entrap users. It exists as a means to locate intentional violations of security policy.

Most audit records are generated in the course of normal work. Even records with ominous sounding names, such as `sat_access_denied`, happen in the course of ordinary activities. Your auditor does not spy on your system activity; he or she guards against an outsider attempting to damage your work.

You do not need to take any action regarding the audit trail. It is maintained by the system and by the auditor at your site. The auditing process is completely transparent to the user.

Chapter 7

Programming in a Trusted Environment

This chapter describes the special requirements of programming in a trusted environment, and lists new system and library calls available under Trusted IRIX/CMW.

Trusted IRIX/CMW conforms to the specifications in POSIX P1003.1eD15.

Sections in this chapter include:

- "Guidelines" on page 66
- "Trusted IRIX/CMW System and Library Calls" on page 66

Guidelines

There are a number of guidelines that anyone who programs in a secure environment should follow:

- In order to simplify your work, do not duplicate the work done by the I&A programs of the Trusted IRIX/CMW system.
- Make sure that all variables are in bounds.
- Reduce global variable usage wherever possible.
- Limit the functionality of each module to only one distinct task.
- Do not create a procedure that circumvents any of the programmatic flow.
- If overrides must be added, document them thoroughly in the code.
- By design and principle, minimize the use of privilege required or permitted by your programs.

Trusted IRIX/CMW System and Library Calls

The following system and library calls are exclusive to Trusted IRIX/CMW. Reference pages exist for each of these calls in reference page sections 2 and 3. Table 8-1 below lists each call and its corresponding action.

Table 7-1 Trusted IRIX/CMW System and Library Calls

System/Library Call	Action
getlabel(2), setlabel(2)	Get or set the MAC label of a file
getplabel(2), setplabel(2)	Get or set the MAC label of a process
recvl(2), recvlfrom(2), recvlmsg(2)	Receive a message and its MAC label from a socket
recvlu(2), recvlufrom(2), recvlumsg(2)	Receive a message, its MAC label, and the UID of the sender from a socket
satctl(2)	Control the collection of audit data
satread(2)	Read a block of audit record data
satwrite(2)	Write a block of audit record data

Table 7-1 (continued) Trusted IRIX/CMW System and Library Calls

System/Library Call	Action
acl_copy_ext(3C)	Copy ACL from system to user space or from user to system space
acl_delete_def_file(3C)	Delete the default ACL for a named directory
acl_dup(3C)	Make a copy of an ACL
acl_free(3C)	Free memory allocated by ACL interface calls
acl_from_text(3C)	Convert a POSIX ACL string to a struct acl or a struct acl to a POSIX ACL string
acl_get_fd(3C)	Get or set the ACL associated with an open file
acl_get_file(3C)	Get or set the ACL for a pathname
acl_size(3C)	Return the size of an ACL
acl_valid(3C)	Validate an ACL
cap_acquire(3C)	Make permitted set capabilities effective or remove effective capabilities
cap_clear(3C)	Clear the fields of a capability
cap_copy_ext(3C)	Copy capability from system to user space or from user to system space
cap_dup(3C)	Make a copy of a capability
cap_free(3C)	Free allocated capability
cap_from_text(3C), cap_to_text, cap_value_to_text	Convert a POSIX capabilities string to internal form, convert capabilities to a POSIX capabilities string, or return the POSIX name for a capability value
cap_get_fd(3C), cap_set_fd	Get or set the capabilities for an open file
cap_get_file(3C), cap_set_file	Get or set the capabilities for a pathname

Table 7-1 (continued) Trusted IRIX/CMW System and Library Calls

System/Library Call	Action
cap_get_flag(3C), cap_set_flag	Get or set the value of a capability flag in a capability
cap_get_proc(3C), cap_set_proc	Get or set process capabilities
cap_init(3C)	Allocate a capability stucture
cap_size(3C)	Return the size of a capability
getspwnam(3)	Get a user's name from the administrative database
getuserinfonam(3), getuserinfouid(3)	Get information about a user.
ia_audit(3)	Create and write an audit record, using *satwrite*
*l*ibperfex(3C), start_counters, read_counters, print_counters	A procedural interface to R10000 counters
mac_cleared(3C), mac_clearedlbl, mac_lowest, mac_lowestlbl	Report on user's clearance
mac_label_devs(3C)	Relabel all listed character devices
mac_dominate(3C)	Compare two MAC labels for dominance relationship
mac_dup(3C)	Produce a duplicate copy of a MAC label
mac_equal(3C)	Compare two MAC labels for the equality relationship
mac_from_text(3C)	Convert an ASCII MAC label string to a binary format MAC label
mac_size(3C)	Get the size of a MAC label
mac_to_text(3)	Convert a binary format MAC label to an ASCII MAC label string
mac_valid(3C)	Test a MAC label for validity
sat_eventtostr(3), sat_strtoevent(3)	Convert an audit event index to or from an audit event string

Table 7-1 (continued) Trusted IRIX/CMW System and Library Calls

System/Library Call	Action
sat_intrp_pathname(3)	Portable interface to interpret *sat_pathname* structs
sat_read_file_info(3), sat_write_file_info, sat_free_file_info	Portable interfaces to read audit file headers
sat_read_header_info(3), sat_free_header_info	Portable interfaces to read audit record headers
sgi_cap_cleared(3C)	Determine whether a user's allowed capabilities are sufficient
sgi_cap_set_from_text(3C)	Set all capabilities from a capabilities string
sgi_getcapabilitybyname(3C)	Get the default and allowed capability sets for a named user

Appendix A

Glossary Of Computer Security Terms

The terms listed in this glossary are used in the trusted systems community.

*-property
A Bell-La Padula security model rule allowing a subject write access to an object only if the security level of the object dominates the security level of the subject. Also called confinement property.

acceptance inspection
The final inspection to determine whether or not a facility or system meets the specified technical and performance standards. This inspection is held immediately after facility and software testing and is the basis for commissioning or accepting the information system.

access
A specific type of interaction between a subject and an object that results in the flow of information from one to the other.

access control
The process of limiting access to the resources of a system only to authorized programs, processes, or other systems (in a network). *See* controlled access and limited access.

access control list
A discretionary access control entity associated with an object, consisting of a list of entries where each entry is an identifier (a user or group of users) coupled with a set of access permissions for that user or group.

access control mechanism
Hardware or software features, operating procedures, management procedures, and various combinations of these designed to detect and prevent unauthorized access and to permit authorized access in an automated system.

access level
The hierarchical portion of the security level used to identify the sensitivity of data and the clearance or authorization of users.

Note: The access level, in conjunction with the nonhierarchical categories, forms the sensitivity label of an object. *See* category, security level, and sensitivity label.

access period

A segment of time, generally expressed on a daily or weekly basis, during which access rights prevail.

access port

A logical or physical identifier that a computer uses to distinguish different tty input/output data streams.

access type

The nature of an access right to a particular device, program, or file (for example, read, write, execute, append, modify, delete, or create).

accountability

The property that enables activities on a system to be traced to individuals who may then be held responsible for their actions.

add-on security

The retrofitting of protection mechanisms, implemented by hardware or software.

administrative security

The management constraints and supplemental controls established to provide an acceptable level of protection for data. Also called procedural security.

administrator

In the trusted system, the administrator is responsible for system administration tasks: filesystem maintenance and repair, account creation, and other miscellaneous administrative duties.

assurance

A measure of confidence that the security features and architecture of an AIS accurately mediate and enforce the security policy.

attack

The act of trying to bypass security controls on a system. An attack may be active, resulting in the alteration of data; or passive, resulting in the release of data.

Note: The fact that an attack is made does not necessarily mean that it will succeed. The degree of success depends on the vulnerability of the system or activity and the effectiveness of existing countermeasures.

audit trail

A chronological record of system activities that is sufficient to enable the reconstruction, review, and examination of the sequence of environments and activities surrounding or leading to an operation, a procedure, or an event in a transaction from its start to final results.

Alternatively, a set of records that collectively provide documentary evidence of processing used to aid in tracing from original transactions forward to related records and reports, or backwards from records and reports to their component source transactions.

auditor

The auditor is an administrator who maintains and examines the system audit trail. This person is responsible for maintaining and rearchiving the information, examining the records for abuse, and customizing the audit record gathering configuration.

authenticate

To verify the identity of a user, device, or other entity in a computer system, often as a prerequisite to allowing access to resources in a system.

Alternatively, to verify the integrity of data that have been stored, transmitted, or otherwise exposed to possible unauthorized modification.

Alternatively, to establish the validity of a claimed identity.

authentication

Verifying the claimed identity of a principal.

authenticator

The means used to confirm the identity or to verify the eligibility of a station, originator, or individual.

Alternatively, a record containing information that can be shown to have been recently generated using the session key known only by the client and server.

authorization

The granting of access rights to a user, program, or process.

Alternatively, the process of determining whether a client may use a service, which objects the client is allowed to access, and the type of access allowed for each.

availability of data

The state when data are in the place needed by the user, at the time the user needs them, and in the form needed by the user.

back door

See trap door.

backup plan

See contingency plan.

bandwidth

A characteristic of a communication channel that is the amount of information that can be passed through it in a given amount of time, usually expressed in bits per second.

Bell-LaPadula model

A formal state transition model of computer security policy that describes a set of access control rules. In this formal model, the entities in a computer system are divided into abstract sets of subjects and objects. The notion of a secure state is defined, and it is proven that each state transition preserves security by moving from secure state to secure state, thereby inductively proving that the system is secure. A system state is defined to be secure if the only permitted access modes of subjects to objects are in accordance with a specific security policy. In order to determine whether or not a specific access mode is allowed, the clearance of a subject is compared to the classification of the object, and a determination is made as to whether the subject is authorized for the specific access mode. *See* star property (*-property) and simple security property.

benign environment

A nonhostile environment that may be protected from external hostile elements by physical, personnel, and procedural security countermeasures.

between-the-lines entry

Unauthorized access obtained by tapping the temporarily inactive tty of a legitimate user. *See* piggyback.

beyond A1

A level of trust defined by the DoD Trusted Computer System Evaluation Criteria (TCSEC) that is beyond the state-of-the-art technology available at the time the criteria were developed. It includes all A1-level features plus additional features not required at the A1 level.

browsing

The act of searching through storage to locate or acquire information without necessarily knowing of the existence or the format of the information being sought.

callback

A procedure for identifying a remote system. In a callback, the host system disconnects the caller and then dials the authorized telephone number of the remote system to reestablish the connection. Synonymous with dial back.

capability
A capability is an attribute of a process that determines whether or not a process has the appropriate privilege to perform a specific action where appropriate privilege is required. Each capability may have associated with it one or more flags. For processes, three flags are always associated with the capability, namely the effective, the permitted, and the inheritable flag. A file may have zero or more of these flags associated with it for a capability. Appropriate privilege is determined solely by a process having a specific capability's effective capability flag set.

category
The non-hierarchical component of the MSEN portion of a security label. That is, a logical division of information that spans hierarchical security levels as a means of increasing the protection of the data and further restricting access to the data. Typical examples would be *Politics*, *Art*, or *Sports*. There can be up to 65,536 different categories on your system.

certification
The technical evaluation of a system's security features that establishes the extent to which a particular computer system's design and implementation meet a set of specified security requirements.

channel
An information transfer path within a system. May also refer to the mechanism by which the path is affected.

ciphertext
The output of an encryption function. Encryption transforms plaintext into ciphertext.

clearance
A security clearance, or "clearance," represents the combination of sensitivity level and categories that you are permitted to access.

client
A process that makes use of a network service, on behalf of a user. Note that in some cases a server may itself be a client of some other server (for example, a print server may be a client of a file server).

closed security environment
An environment in which both of the following conditions hold true: (1) Application developers (including maintainers) have sufficient clearances and authorizations to provide an acceptable presumption that they have not introduced malicious logic. (2) Configuration control provides sufficient assurance that applications and the equipment are protected against the introduction of malicious logic before and during the operation of system applications.

communications security
Measures taken to deny unauthorized persons information derived from telecommunications of the U.S. government concerning national security, and to ensure the authenticity of such telecommunications. Communications security includes cryptosecurity, transmission security, and physical security of communications security material and information.

compromise
A violation of the security policy of a system such that unauthorized disclosure of sensitive information may have occurred.

compromising emanations
Unintentional data-related or intelligence-bearing signals that, if intercepted and analyzed, disclose the information transmission received, handled, or otherwise processed by any information processing equipment.

computer abuse
The misuse, alteration, disruption or destruction of data processing resources. The key aspects are that it is intentional and improper.

computer cryptography
The use of a crypto-algorithm in a computer, microprocessor, or microcomputer to perform encryption or decryption in order to protect information or to authenticate users, sources, or information.

computer fraud
Computer-related crimes involving deliberate misrepresentation, alteration, or disclosure of data in order to obtain something of value (usually for monetary gain). A computer system must have been involved in the perpetration or cover-up of the act or series of acts. A computer system might have been involved through improper manipulation of input data; output or results; applications programs; data files; computer operations; communications; or computer hardware, systems software, or firmware.

COMSEC
Refers to communications security.

concealment system
A method of achieving confidentiality in which sensitive information is hidden by embedding it in irrelevant data.

confidentiality
The concept of holding sensitive data in confidence, limited to an appropriate set of individuals or organizations.

configuration control

The process of controlling modifications to the system's hardware, firmware, software, and documentation that provides sufficient assurance that the system is protected against the introduction of improper modifications before, during, and after system implementation.

configuration management

The management of security features and assurances through control of changes made to a system's hardware, software, firmware, documentation, test, test fixtures, and test documentation throughout the development and operational life of the system.

configuration range

The evaluation of a computer system by the NCSC (National Computer Security Center) is typically performed on a set of computer systems manufactured by the evaluatee rather than on just one particular computer system model. Due to the complexity of the evaluation process, it is common that only a closely related subset of the evaluatee's computer system product line be evaluated. The exact definition of the set of computer systems that is being evaluated is called the configuration range. The definition is exact. For example, part numbers of cables that connect keyboards to the system are part of the definition, and use of even a keyboard cable with a part number not in the configuration range will cause the evaluation not to be valid for that system. It is important to remember that the whole computer system is being evaluated, not just the software.

confinement The prevention of the leaking of sensitive data from a program.

confinement channel

See covert channel.

contamination The intermixing of data at different sensitivity and need-to-know levels. The lower-level data is said to be contaminated by the higher level data; thus, the contaminating (higher level) data may not receive the required level of protection.

contingency plan

A plan for emergency response, backup operations, and post-disaster recovery maintained by an activity as a part of its security program that ensures the availability of critical resources and facilitates the continuity of operations in an emergency situation. Also called disaster plan and emergency plan.

control zone
The space (expressed in feet of radius, surrounding equipment processing sensitive information) that is under sufficient physical and technical control to preclude an unauthorized entry or compromise.

controlled access
See access control.

controlled sharing
The condition that exists when access control is applied to all users and components of a system.

cost-risk analysis
The assessment of the costs of providing data protection for a system versus the cost of losing or compromising the data.

countermeasure
Any action, device, procedure, technique, or other measure that reduces the vulnerability of or threat to a system.

covert channel
A communications channel that allows two cooperating processes to transfer information in a manner that violates the system's security policy. Synonymous with confinement channel.

Alternatively, a communication channel that allows a process to transfer information in a manner that violates the system's security policy. *See also* covert storage channel and covert timing channel.

covert storage channel
A covert channel that involves the direct or indirect writing of a storage location by one process and the direct or indirect reading of the storage location by another process. Covert storage channels typically involve a finite resource (for example, sectors on a disk) that is shared by two subjects at different security levels.

covert timing channel
A covert channel in which one process signals information to another by modulating its own use of system resources (for example, CPU time) in such a way that this manipulation affects the real response time observed by the second process.

crypto-algorithm
A well-defined procedure or sequence of rules or steps used to produce a key stream or cipher text from plain text and vice versa.

cryptography
The principles, means and methods for rendering information unintelligible, and for restoring encrypted information to intelligible form.

cryptosecurity
The security or protection resulting from the proper use of technically sound cryptosystems.

DAC
Discretionary Access Control.

data
Information with a specific physical representation.

data flow control
See information flow control.

data integrity
The requirement that data meet an a priori expectation of quality.

Alternatively, the state that exists when computerized data is the same as that in the source documents and has not been exposed to accidental or malicious alteration or destruction.

data security
The protection of data from unauthorized (accidental or intentional) modification, destruction, or disclosure.

dedicated security mode
See modes of operation.

default classification
A temporary classification reflecting the highest classification being processed in a system. The default classification is included in the caution statement affixed to the object.

degauss
To reduce magnetic flux density to zero by applying a reverse magnetizing field.

degausser
An electrical device that can generate a magnetic field for the purpose of degaussing magnetic storage media.

Degausser Products List
A list of commercially produced degaussers that meet National Security Agency specifications. This list is included in the NSA Information Systems Security Products and Services Catalogue, and is available through the Government Printing Office.

denial of service

Any action or series of actions that prevent any part of a system from functioning in accordance with its intended purpose. This includes any action that causes unauthorized destruction, modification, or delay of service. Synonymous with interdiction.

dial back *See* callback.

dial up The service whereby a computer can use the telephone to initiate and effect communication with a computer.

disaster plan *See* contingency plan.

Discretionary Access Control

A means of restricting access to objects based on the identity and the need of the user, process and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject. *See also* Mandatory Access Control.

division The non-hierarchical component of the MINT part of the security label. This is the integrity part of the security label. The division component is very similar to the category component of the MSEN part of the security label. There can be up to 65,536 different divisions on your system. Typical examples of a division might be *Prose*, *Poetry*, *Verse*.

DoD Trusted Computer System Evaluation Criteria

A document published by the National Computer Security Center containing a uniform set of basic requirements and evaluation classes for assessing degrees of assurance in the effectiveness of hardware and software security controls built into systems. These criteria are intended for use in the design and evaluation of systems that process and store sensitive or classified data. This document is Government Standard DoD 5200.28-STD and is frequently referred to as “The Criteria” or “The Orange Book”.

domain The unique context (for example, access control parameters) in which a program is operating; in effect, the set of objects that a subject has the ability to access. *See* process and subject.

Alternatively, the set of objects that a subject has the ability to access.

dominate

Access to a file or resource under Mandatory Access Control is determined according to “domination.” You can view a file only if your process label dominates the label of the file. One label (*high*) dominates another label (*low*) if all four of the following conditions are true:

- The Mandatory Sensitivity of *high* is greater than or equal to *low,*
- The set of Mandatory Sensitivity categories of *high* is identical to or a strict superset of the categories of *low,*
- The Mandatory Integrity requirement of *high* is less than or equal to the Integrity grade of *low,*
- The set of Mandatory Integrity divisions of *high* is identical to or a strict superset of the divisions of *low.*

emanations

See compromising emanations.

embedded system

A system that performs or controls a function, either in whole or in part, as an integral element of a larger system or subsystem.

emergency plan

See contingency plan.

emission security

The protection resulting from all measures taken to deny unauthorized persons information of value that might be derived from intercept and from an analysis of compromising emanations from systems.

end-to-end encryption

The protection of information passed in a telecommunications system by cryptographic means, from point of origin to point of destination.

Alternatively, protection of traffic in a communications network by encrypting it at the source and decrypting it at the destination so that all nodes it passes through remain ignorant of its actual content.

entrapment

The deliberate planting of apparent flaws in a system for the purpose of detecting attempted penetrations.

environment

The aggregate of external procedures, conditions, and objects that affect the development, operation, and maintenance of a system.

EPL

The Evaluated Products List

erasure A process by which a signal recorded on magnetic media is removed. Erasure is accomplished in two ways: (l) alternating current erasure, in which the information is destroyed by applying an alternating high and low magnetic field to the media; or (2) direct current erasure, in which the media are saturated by applying a unidirectional magnetic field.

Evaluated Products List

A list of equipments, hardware, software, and firmware that have been evaluated against and found to be technically compliant with, at a particular level of trust, the DoD TCSEC by the NCSC. The EPL is included in the National Security Agency Information Systems Security Products and Services Catalogue, which is available through the Government Printing Office.

evaluation criteria

Trusted IRIX/CMW meets the requirements specified for the Trusted Computer System Evaluation Criteria (TCSEC). The U.S. government specifies a set of criteria that trusted systems must meet to be evaluated successfully. A trusted system must offer a number of specific security features and must demonstrate that it can be maintained and distributed in a trusted fashion.

executive state One of several states in which a system may operate and the only one in which certain privileged instructions may be executed. Such instructions cannot be executed when the system is operating in other (for example, user) states. Synonymous with supervisor state.

exploitable channel

Any information channel that is usable or detectable by subjects external to the trusted computing base whose purpose is to violate the security policy of the system. *See also* covert channel.

Alternatively, any channel that is usable or detectable by subjects external to the Trusted Computing Base.

fail safe Pertaining to the automatic protection of programs or processing systems to maintain safety when a hardware or software failure is detected in a system.

fail soft Pertaining to the selective termination of affected nonessential processing when a hardware or software failure is detected in a system.

failure access An unauthorized and usually inadvertent access to data resulting from a hardware or software failure in the system.

failure control
The methodology used to detect and provide fail-safe or fail-soft recovery from hardware and software failures in a system.

fault
A condition that causes a device or system component to fail to perform in a required manner.

fetch protection
A system-provided restriction to prevent a program from accessing data in another user's segment of storage.

file protection
The aggregate of all processes and procedures in a system designed to inhibit unauthorized access, contamination, or elimination of a file.

file security
The means by which access to computer files is limited to authorized users only.

flaw
An error of commission, omission, or oversight in a system that allows protection mechanisms to be bypassed.

flaw hypothesis methodology
A systems analysis and penetration technique in which specifications and documentation for the system are analyzed and flaws in the system are hypothesized. The list of hypothesized flaws is then prioritized on the basis of the estimated probability that a flaw exists and, assuming a flaw does exist, on the ease of exploiting it, and on the extent of control or compromise it would provide. The prioritized list is used to direct a penetration attack against the system.

formal access approval
Documented approval by a data owner to allow access to a particular category of information.

formal proof
A complete and convincing mathematical argument, presenting the full logical justification for each proof step, for the truth of a theorem or set of theorems. The formal verification process uses formal proofs to show the truth of certain properties of formal specification and for showing that computer programs satisfy their specifications.

formal security policy model

A mathematically precise statement of a security policy. To be precise, such a model must represent the initial state of a system, the way in which the system progresses from one state to another, and a definition of a secure state of the system. To be acceptable as a basis for a Trusted Computing Base, the model must be supported by a formal proof that if the initial state of the system satisfies the definition of a "secure" state and if all assumptions required by the model hold, then all future states of the system will be secure. Some formal modeling techniques include: state transition models, temporal logic models, denotational semantics models, algebraic specification models. An example is the model described by Bell and LaPadula in [Bell, D. E. and LaPadula, L. J. Secure Computer System: Unified Exposition and Multics Interpretation, MTR-2997 Rev. 1, MITRE Corp., Bedford, Mass., March 1976]. *See also* Bell-LaPadula model and security policy model.

formal verification

The process of using formal proofs to demonstrate the consistency between a formal specification of a system and a formal security policy model (design verification) or between the formal specification and its high-level program implementation (implementation verification).

front-end security filter

A security filter, which could be implemented in hardware or software, that is logically separated from the remainder of the system to protect the system's integrity.

Alternatively, a process that is invoked to process data according to a specified security policy prior to releasing the data outside the processing environment or upon receiving data from an external source.

functional testing

The segment of security testing in which the advertised security features of the system are tested, under operational conditions, for correct operation.

grade — The hierarchical component of the MINT part of the security label. This is the representation of the integrity level of an object or the integrity requirement of a subject. The higher the value, the higher the integrity level or requirement. Typical examples of grade are as follows:

- Best--This integrity rating is reserved for the Trusted Computing Base. Administrative accounts such as root require this level of integrity.
- Good--Free from viruses, worms, and so on.
- Poor-- Software obtained from unknown persons.

granularity — An expression of the relative size of a data object; for example, protection at the file level is considered coarse granularity, whereas protection at field level is considered to be of a finer granularity.

Alternatively, the relative fineness or coarseness by which a mechanism can be adjusted. The phrase “the granularity of a single user” means the access control mechanism can be adjusted to include or exclude any single user.

guard — A processor that provides a filter between two disparate systems operating at different security levels or between a user process and a database to filter out data that the user is not authorized to access.

handshaking procedure — A dialogue between two entities (for example, a user and a computer, a computer and another computer, or a program and another program) for the purpose of identifying and authenticating the entities to one another.

host to front-end protocol — A set of conventions governing the format and control of data that are passed from a host to a front-end machine.

I&A — Identification and Authentication.

identification — The process that enables recognition of an entity by a system, generally by the use of unique machine-readable user names.

Identification and Authentication
I&A is the process of determining (with some level of confidence) the true identity of a user. The identification process usually requires both a user name and a password. The authentication part of the process is the underlying logic that the *login* and *su* programs go through in validating this password and username.

impersonating *See* spoofing.

incomplete parameter checking
A system design flaw that results when not all parameters have been fully anticipated for accuracy and consistency, thus making the system vulnerable to penetration.

individual accountability
The ability to associate positively the identity of a user with the time, method, and degree of access to a system.

information flow control
A procedure to ensure that information transfers within a system are not made from a higher security level object to an object of a lower security level. *See* covert channel, simple security property, and star property (*-property). Synonymous with data flow control.

Information Systems Security Products and Services Catalogue
A catalogue issued quarterly by the National Security Agency that incorporates the DPL, EPL, ETL, PPL and other security product and service lists. This catalogue is available through the U.S. Government Printing Office, Washington, DC (202) 1202) 783-3238.

instance The name often given to the second component of a principal identifier, or a particular principal from a group of related principals. In the latter usage, the instances are often created to partition permission for users. For example, a user might have a “normal” instance and a “root” instance (which has different privileges) to impose a naming convention on service key names. For an example of a particular service, the instances identifies the host machines on which that service is provided and the principal identifier of the server.

integrity

In secure systems, the term “integrity” refers to the relative level of trust a user can place in using a system resource. A program obtained from a public-access bulletin board is of much lower integrity than one purchased from a reputable vendor. This program is in turn of much lower integrity than a program shipped as part of a trusted system. *See* Mandatory Integrity.

Alternatively, sound, unimpaired, or perfect condition.

integrity label-- MINT

One half of the MAC label. Represents the measure of trust a user can put in a system resource. *See* MSEN and sensitivity label.

interdiction

See denial of service.

internal security controls

Hardware, firmware, and software features within a system that restrict access to resources (hardware, software, and data) to authorized subjects only (persons, programs, or devices).

isolation

The containment of subjects and objects in a system in such a way that they are separated from one another, as well as from the protection controls of the operating system.

lattice

A partially ordered set for which every pair of elements has a greatest lower bound and a least upper bound.

least privilege

The principle that requires that each subject be granted the most restrictive set of privileges needed for the performance of authorized tasks. The application of this principle limits the damage that can result from accident, error, or unauthorized use.

limited access

See access control.

lock-and-key protection system

A protection system that involves matching a key or password with a specific access requirement.

logic bomb

A resident computer program that triggers the perpetration of an unauthorized act when particular states of the system are realized.

login-spoofing program

This term refers to any program that represents itself as a login program in order to steal your password. For example, a spoofing program might print the UNIX login banner on an unattended system and wait for input from the user. The user dutifully types in the user name, and the program prompts for the password, turning off character echo. After storing away the user's password, the program reports that the password is incorrect and exits, which causes the real login program to be started on the system. The user then logs in, mistakenly assuming that he or she previously mistyped the name or password, and starts a session.

loophole

An error of omission or oversight in software or hardware that permits circumventing the system security policy.

MAC

Mandatory Access Control. *See* Mandatory Access Control.

MAC label

A MAC label is comprised of two halves: the sensitivity label (MSEN) and the integrity label (MINT). A typical MAC label would be msenhigh/mintlow. This would represent an object with a highly sensitive topic, but with a relatively low level of integrity (perhaps obtained from a questionable source).

magnetic remanence

A measure of the magnetic flux density remaining after removal of the applied magnetic force. Refers to any data remaining on magnetic storage media after removal of the power.

maintenance hook

Special instructions in software to allow easy maintenance and additional feature development. These are not clearly defined during access for design specification. Hooks frequently allow entry into the code at unusual points or without the usual checks, so they are a serious security risk if they are not removed before live implementation. Maintenance hooks are special types of trap doors.

malicious logic

Hardware, software, or firmware that is intentionally included in a system for an unauthorized purpose; for example, a Trojan Horse virus.

Mandatory Access Control

MAC is a means of restricting access to objects based on the sensitivity and integrity (as represented by a label) of the information contained in the objects and the formal authorization (that is, clearance) of subjects to access information of such sensitivity and integrity. *See also* Discretionary Access Control.

Mandatory Integrity

A means of restricting access to objects based on the integrity (as represented by a label) of the information contained in the objects and the subjects. Integrity is necessary to identify the Trusted IRIX/CMW TCB. In order to do so, some mechanism for restricting what programs may be executed by the superuser, auditor, and any other trusted users must be implemented. The Mandatory Integrity (MINT) component of the security label provides TCB isolation by denying access to programs that have not been sufficiently analyzed (or have been and are deemed untrustworthy to users with high integrity requirements). The MINT mechanism allows only those processes whose integrity labels are dominated by an object read or execute access to it. Additionally, a process may write only to an object with the same integrity. The MINT mechanism is very similar to the MSEN mechanism in having 256 hierarchical levels (the grades) and 65,536 non-hierarchical components (the divisions).

Mandatory Sensitivity

The label of every subject and object on the system indicates a level of security clearance. Access to an object by a subject is based on their relative levels of clearance. A user will not even be aware of the existence of objects that are at a higher level of sensitivity. A sensitivity label (MSEN) is comprised of a type (for example, msenhigh) and a category (for example, Politics, Sports).

masquerading *See* spoofing.

mimicking *See* spoofing.

MINT Mandatory Integrity. *See* Mandatory Integrity.

multilevel device

A device that is used in a manner that permits it to simultaneously process data of two or more security levels without risk of compromise. To accomplish this, sensitivity labels are normally stored on the same physical medium and in the same form (that is, machine-readable or human-readable) as the data being processed.

multilevel secure

A class of system containing information with different sensitivities that simultaneously permits access by users with different security clearances and needs-to-know, but prevents users from obtaining access to information for which they lack authorization.

multilevel security mode

See modes of operation.

multiple access rights terminal

A system or port that may be used by more than one class of users; for example, users with different access rights to data.

multiuser mode of operation

A mode of operation designed for systems that process sensitive unclassified information in which users may not have a need-to-know for all information processed in the system. This mode is also for microcomputers processing sensitive unclassified information that cannot meet the requirements of the standalone mode of operation.

mutually suspicious

The state that exists between interacting processes (subsystems or programs) in which neither process can expect the other process to function securely with respect to some property.

National Computer Security Center

Originally named the DoD Computer Security Center, the NCSC is responsible for encouraging the widespread availability of trusted computer systems throughout the Federal Government.

National Security Decision Directive 145

Signed by President Reagan on 17 September 1984, this directive is entitled “National Policy on Telecommunications and Automated Information Systems Security.” It provides initial objectives, policies, and an organizational structure to guide the conduct of national activities toward safeguarding systems that process, store, or communicate sensitive information; establishes a mechanism for policy development; and assigns implementation responsibilities.

National Telecommunications and Information System Security Directives

NTISS Directives establish national-level decisions relating to NTISS policies, plans, programs, systems, or organizational delegations of authority. NTISSDs are promulgated by the Executive Agent of the Government for Telecommunications and Information Systems Security, or by the Chairman of the NTISSC when so delegated by the Executive Agent. NTISSDs are binding upon all federal departments and agencies.

National Telecommunications and Information Systems Security Advisory Memoranda Instructions

NTISS Advisory Memoranda and Instructions provide advice, assistance, or information of general interest on telecommunications and systems security to all applicable federal departments and agencies. NTISSAMs/NTISSIs are promulgated by the National Manager for Telecommunications and Automated Information Systems Security and are recommendatory.

NCSC National Computer Security Center.

need-to-know The necessity for access to, knowledge of, or possession of specific information required to carry out official duties.

network front end

A device that implements the necessary network protocols, including security-related protocols, to allow a computer system to be attached to a network.

NSDD 145 See National Security Decision Directive 145.

NTISSC National Telecommunications and Information Systems

object A passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory trees, and programs, as well as bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, and network nodes.

object reuse The reassignment and reuse of a storage medium (for example, page frame, disk sector, magnetic tape) that once contained one or more objects. To be securely reused and assigned to a new subject, storage media must contain no residual data (magnetic remanence) from the object(s) previously contained in the media.

open security environment

An environment that includes those systems in which at least one of the following conditions holds true: (1) Application developers (including maintainers) do not have sufficient clearance or authorization to provide an acceptable presumption that they have not introduced malicious logic. (2) Configuration control does not provide sufficient assurance that applications are protected against the introduction of malicious logic prior to and during the operation of system applications.

Operations Security

An analytical process by which the U.S. Government and its supporting contractors can deny to potential adversaries information about capabilities and intentions by identifying, controlling, and protecting evidence of the planning and execution of sensitive activities and operations.

OPSEC

Operations Security. *See* Operations Security.

Orange Book

Alternate name for DoD Trusted Computer Security Evaluation Criteria.

output

Information that has been exported by a TCB.

overt channel

A path within a computer system or network that is designed for the authorized transfer of data. *See also* covert channel.

overwrite procedure

A stimulation to change the state of a bit followed by a known pattern. *See also* magnetic remanence.

password

A protected, private character string used to authenticate an identity.

penetration

The successful act of bypassing the security mechanisms of a system.

penetration signature

The characteristics or identifying marks that may be produced by a penetration.

penetration study

A study to determine the feasibility and methods for defeating controls of a system.

penetration testing

The portion of security testing in which the evaluators attempt to circumvent the security features of a system. The evaluators may be assumed to use all system design and implementation documentation, which may include listings of system source code, manuals, and circuit diagrams. The evaluators work under the same constraints applied to ordinary users.

periods processing

The processing of various levels of sensitive information at distinctly different times. Under periods processing, the system must be purged of all information from one processing period before transitioning to the next when there are different users with differing authorizations.

permissions A description of the type of authorized interactions a subject can have with an object. Examples include read, write, execute, add, modify, and delete.

personnel security

The procedures established to ensure that all personnel who have access to sensitive information have the required authority as well as appropriate clearances.

physical security

The application of physical barriers and control procedures as preventive measures or countermeasures against threats to resources and sensitive information.

piggyback Gaining unauthorized access to a system via another user's legitimate connection. *See* between-the-lines entry.

plaintext The input to an encryption function or the output of a decryption function. Decryption transforms ciphertext into plaintext.

Preferred Products List

A list of commercially produced equipments that meet requirements prescribed by the National Security Agency. This list is included in the NSA Information Systems Security Products and Services Catalogue, issued quarterly and available through the Government Printing Office.

principal A uniquely named client or server instance that participates in a network communication.

principal identifier

The name used to uniquely identify each different principal.

print suppression
Eliminating the displaying of characters in order to preserve their secrecy; for example, not displaying the characters of a password as it is keyed in.

privileged instructions
A set of instructions (for example, interrupt handling or special computer instructions) to control features (such as storage protection features) that are generally executable only when the automated system is operating in the executive state.

procedural security
See administrative security.

process
A program in execution. It is completely characterized by a single current execution point (represented by the machine state) and address space.

protection-critical portions of the TCB
Those portions of the TCB whose normal function is to deal with the control of access between subjects and objects. Their correct operation is essential to the protection of the data on the system.

protection philosophy
An informal description of the overall design of a system that delineates each of the protection mechanisms employed. A combination, appropriate to the evaluation class, of formal and informal techniques is used to show that the mechanisms are adequate to enforce the security policy.

protection ring
One of a hierarchy of privileged modes of a system that gives certain access rights to user programs and processes authorized to operate in a given mode.

protocols
A set of rules and formats, semantic and syntactic, that permits entities to exchange information.

pseudo-flaw
An apparent loophole deliberately implanted in an operating system program as a trap for intruders.

Public Law 100-235

Also known as the Computer Security Act of 1987, this law creates a means for establishing minimum acceptable security practices for improving the security and privacy of sensitive information in federal computer systems. This law assigns to the National Institute of Standards and Technology responsibility for developing standards and guidelines for federal computer systems processing unclassified data. The law also requires establishment of security plans by all operators of federal computer systems that contain sensitive information.

rainbow series

The informal name given to a set of books published by the NCSC that deal with computer security. The books are published with covers in different colors, hence the term "rainbow." The most used book in the rainbow series is the Orange Book, the DoD Trusted Computer System Evaluation Criteria.

read

A fundamental operation that results only in the flow of information from an object to a subject.

read access

Permission to read information.

recovery procedures

The actions necessary to restore a system's computational capability and data files after a system failure.

reference monitor concept

An access-control concept that refers to an abstract machine that mediates all accesses to objects by subjects.

reference validation mechanism

An implementation of the reference monitor concept. A security kernel is a type of reference validation mechanism.

reliability

The probability of a given system performing its mission adequately for a specified period of time under the expected operating conditions.

residual risk

The portion of risk that remains after security measures have been applied.

residue

Data left in storage after processing operations are complete, but before degaussing or rewriting has taken place.

resource encapsulation

The process of ensuring that a resource not be directly accessible by a subject, but that it be protected so that the reference monitor can properly mediate accesses to it.

restricted area Any area to which access is subject to special restrictions or controls for reasons of security or safeguarding of property or material.

risk The probability that a particular threat will exploit a particular vulnerability of the system.

risk analysis The process of identifying security risks, determining their magnitude, and identifying areas needing safeguards. Risk analysis is a part of risk management. Synonymous with risk assessment.

risk assessment S*ee* risk analysis.

risk index The disparity between the minimum clearance or authorization of system users and the maximum sensitivity (for example, classification and categories) of data processed by a system.

risk management

The total process of identifying, controlling, and eliminating or minimizing uncertain events that may affect system resources. It includes risk analysis, cost benefit analysis, selection, implementation and test, security evaluation of safeguards, and overall security review.

RM Plan The Rating Maintenance Plan (RM-Plan) is a living document that describes the policies which govern modifications to the Trusted IRIX/CMW system and the procedures used to implement these policies. It describes in detail the initial contents of the system, and how each component of the Trusted Computing Base was approved for inclusion. The procedures for making changes to the system for future releases are defined. The change request mechanisms for new features, performance enhancements, and field-detected security failures are described. The methods by which changes are tracked are defined in the RM-Plan. Source code control, document control, the product naming scheme, and methods for identification of changes to the RM-Plan itself are described. Evidence supporting the validity and necessity of changes is maintained.

safeguards *See* security safeguards.

scavenging Searching through object residue to acquire unauthorized data.

seal To encipher a record containing several fields in such a way that the fields cannot be individually replaced without either knowledge of the encryption key or leaving evidence of tampering.

secure configuration management

The set of procedures appropriate for controlling changes to a system's hardware and software structure for the purpose of ensuring that changes will not lead to violations of the system's security policy.

secure state A condition in which no subject can access any object in an unauthorized manner.

secure subsystem

A subsystem that contains its own implementation of the reference monitor concept for those resources it controls. However, the secure subsystem must depend on other controls and the base operating system for the control of subjects and the more primitive system objects.

Security Administration Guide

This document describes the administration of the security features of Trusted IRIX/CMW. Instructions are provided on planning and administering a trusted system, managing Mandatory Access Control, Auditing, and Identification and Authentication facilities. Also, the document covers printing and use of magnetic media in a trusted environment.

The NCSC requires this document as part of the evaluation materials. The NCSC name for this kind of document is a “Trusted Facilities Manual.” The *Trusted IRIX/CMW Security Administration Guide* is the Trusted Facilities Manual for Trusted IRIX/CMW.

security critical mechanisms

Those security mechanisms whose correct operation is necessary to ensure that the security policy is enforced.

security evaluation
An evaluation done to assess the degree of trust that can be placed in systems for the secure handling of sensitive information. One type, a product evaluation, is an evaluation performed on the hardware and software features and assurances of a computer product from a perspective that excludes the application environment. The other type, a system evaluation, is done for the purpose of assessing a system's security safeguards with respect to a specific operational mission and is a major step in the certification and accreditation process.

security fault analysis
A security analysis, usually performed on hardware at gate level, to determine the security properties of a device when a hardware fault is encountered.

security features
The security-relevant functions, mechanisms, and characteristics of system hardware and software. Security features are a subset of system security safeguards.

Security Features User's Guide
This document exists to describe in layman's terms the user visible portion of the security features of the Trusted IRIX/CMW operating system. This book describes for the user specific methods for effectively using the system. It also describes what the user is not allowed to do and what actions the user should take when faced with a denial of service.

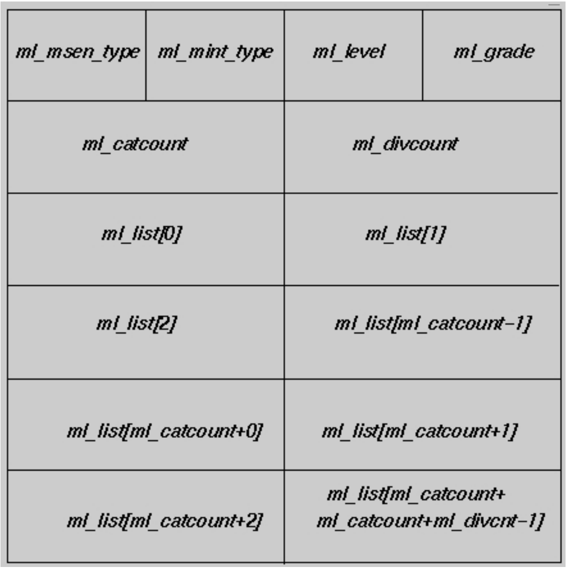
security filter
A trusted subsystem that enforces a security policy on the data that pass through it.

security flaw
An error of commission or omission in a system that may allow protection mechanisms to be bypassed.

security flow analysis
A security analysis performed on a formal system specification that locates potential flows of information within the system.

security kernel
The hardware, firmware, and software elements of a TCB that implement the reference monitor concept. It must mediate all accesses, be protected from modification, and be verifiable as correct.

security label The data structure used to associate a security clearance or classification to each subject and object in Trusted IRIX/CMW. The structure of a security label is shown in Figure A-1.



<table>
<tr><td>ml_msen_type</td><td>ml_mint_type</td><td>ml_level</td><td>ml_grade</td></tr>
<tr><td colspan="2">ml_catcount</td><td colspan="2">ml_divcount</td></tr>
<tr><td colspan="2">ml_list[0]</td><td colspan="2">ml_list[1]</td></tr>
<tr><td colspan="2">ml_list[2]</td><td colspan="2">ml_list[ml_catcount-1]</td></tr>
<tr><td colspan="2">ml_list[ml_catcount+0]</td><td colspan="2">ml_list[ml_catcount+1]</td></tr>
<tr><td colspan="2">ml_list[ml_catcount+2]</td><td colspan="2">ml_list[ml_catcount+
ml_catcount+ml_divcnt-1]</td></tr>
</table>

Figure A-1 Data Structure of a Security Label

security measures
Elements of software, firmware, hardware, or procedures that are included in a system for the satisfaction of security specifications.

security perimeter
The boundary where security controls are in effect to protect assets.

security policy The set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information.

security policy model
A formal presentation of the security policy enforced by the system. It must identify the set of rules and practices that regulate how a system manages, protects, and distributes sensitive information. *See* Bell-La Padula model and formal security policy model.

security range The highest and lowest security levels that are permitted in or on a system, system component, subsystem, or network.

security requirements

The types and levels of protection necessary for equipment, data, information, applications, and facilities to meet security policy.

security requirements baseline

A description of minimum requirements necessary for a system to maintain an acceptable level of security.

security safeguards

The protective measures and controls that are prescribed to meet the security requirements specified for a system. Those safeguards may include but are not necessarily limited to hardware and software security features, operating procedures, accountability procedures, access and distribution controls, management constraints, personnel security, and physical structures, areas, and devices. Also called safeguards.

security specifications

A detailed description of the safeguards required to protect a system.

security test and evaluation

An examination and analysis of the security safeguards of a system as they have been applied in an operational environment to determine the security posture of the system.

security testing

A process used to determine that the security features of a system are implemented as desired. This includes hands-on functional testing, penetration testing, and verification.

sensitive information

Any information, whose loss, misuse, modification of, or unauthorized access to, could affect the national interest or the conduct of Federal programs, or the privacy to which individuals are entitled under Section 552a of Title 5, U.S. Code, but that has not been specifically authorized under criteria established by an Executive order or an act of Congress to be kept classified in the interest of national defense or foreign policy.

sensitivity

In secure systems, sensitivity is a measure of the risk associated with the disclosure of the data in question. A map of a foreign city (UNCLASSIFIED) is less sensitive than the map of a foreign military base (SECRET) which is in turn less sensitive than the name of the asset who provided the maps (TOP SECRET).

sensitivity label

One half of the MAC label. Where the MINT half represents the degree of confidence a user may have in the integrity of a system resource, the sensitivity label is a relative representation of the degree of risk associated with the disclosure of the data in question. Sensitivity labels are used by the TCB as the basis for mandatory access control decisions.

sensitivity level

The sensitivity level is the hierarchical portion of the sensitivity label. *See* sensitivity label.

server — A particular Principal that provides a resource to network clients.

service — A resource provided to network clients; often provided by more than one server (for example, remote file service).

session key — A temporary encryption key used between two principals, with a lifetime limited to the duration of a single communications “session.

”SFUG — *See* Security Features User's Guide.

simple security condition

See simple security property.

simple security property

A Bell-La Padula security model rule allowing a subject read access to an object only if the security level of the subject dominates the security level of the object. Synonymous with simple security condition.

single-level device

An automated information systems device that is used to process data of a single security level at any one time. Because the device need not be trusted to separate data of different security levels, sensitivity labels do not have to be stored with the data being processed.

software security

General purpose (executive, utility or software development tools) and applications programs or routines that protect data handled by a system.

software system test and evaluation process

A process that plans, develops and documents the quantitative demonstration of the fulfillment of all baseline functional performance, operational, and interface requirements.

spoofing — An attempt to gain access to a system by posing as an authorized user. Synonymous with impersonating, masquerading or mimicking.

standalone shared system
A system that is physically and electrically isolated from all other systems, and is intended to be used by more than one person, either simultaneously (for example, a system with multiple monitors) or serially, with data belonging to one user remaining available to the system while another user is using the system (for example, a personal computer with nonremovable storage media such as a hard disk).

standalone single-user system
A system that is physically and electrically isolated from all other systems, and is intended to be used by one person at a time, with no data belonging to other users remaining in the system (for example, a personal computer with removable storage media such as a floppy disk).

star property
See *-property.

state variable
A variable that represents either the state of the system or the state of some system resource.

storage object
An object that supports both read and write accesses.

STS
Subcommittee on Telecommunications Security of NTISSC

Subcommittee on Automated Information Systems Security
NSDD-l 45 authorizes and directs the establishment, under the NTISSC, of a permanent Subcommittee on Automated Information Systems Security. The SAISS is composed of one voting member from each organization represented on the NTISSC.

Subcommittee on Telecommunications Security
NSDD-145 authorizes and directs the establishment, under the NTISSC, of a permanent Subcommittee on Telecommunications Security. The STS is composed of one voting member from each organization represented on the NTISSC.

subject
An active entity, generally in the form of a person, process, or device, that causes information to flow among objects or changes the system state. Technically, a process/domain pair.

subject security level
A subject's security level is equal to the security level of the objects to which it has both read and write access. A subject's security level must always be dominated by the clearance of the user with which the subject is associated.

supervisor state

See executive state.

System Call Security Analysis

A document that describes the security policies, both discretionary and mandatory, enforced by each of the Trusted IRIX/CMW system calls. For each system call the differences in behavior between the superuser and normal users, if any, are described. The object reuse policies are discussed. This document is the heart of the security policy description in that it describes which interfaces to the Trusted Computing Base are affected and implement the system security policy. It is explicit and definitive.

system integrity

The quality that a system has when it performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

system low

The lowest security level supported by a system at a particular time or in a particular environment.

Systems Security Steering Group

The senior government body established by NSDD-145 to provide top-level review and policy guidance for the telecommunications security and automated information systems security activities of the U.S. Government. This group is chaired by the Assistant to the President for National Security Affairs and consists of the Secretary of State, Secretary of Treasury, the Secretary of Defense, the Attorney General, the Director of the Office of Management and Budget, and the Director of Central Intelligence.

tampering

An unauthorized modification that alters the proper functioning of a piece of equipment or system in a manner that degrades the security or functionality it provides.

TCB

Trusted Computing Base. *See* Trusted Computing Base.

TCSEC

DoD Trusted Computer System Evaluation Criteria.

technical attack

An attack that can be perpetrated by circumventing or nullifying hardware and software protection mechanisms, rather than by subverting system personnel or other users.

technical vulnerability
A hardware, firmware, communication, or software flaw that leaves a computer processing system open for potential exploitation, either externally or internally, thereby resulting in risk for the owner, user, or manager of the system.

Test Plan
A single document describes the overall planning for testing Trusted IRIX/CMW. This document discusses the documentation plan for testing, the design goal, software requirements, general testing requirements, testing strategies, approaches, methods, hardware resources, software resources, personnel resources, schedules, and milestones.

TFM
Trusted Facilities Manual. *See also* Security Administration Guide.

threat
Any circumstance or event with the potential to cause harm to a system in the form of destruction, disclosure, modification of data, and/or denial of service.

threat agent
A method used to exploit a vulnerability in a system, operation, or facility.

threat analysis
The examination of all actions and events that might adversely affect a system or operation.

threat monitoring
The analysis, assessment, and review of audit trails and other data collected for the purpose of searching out system events that may constitute violations or attempted violations of system security.

time-dependent password
A password that is valid only at a certain time of day or during a specified interval of time.

top-level specification
A nonprocedural description of system behavior at the most abstract level; typically, a functional specification that omits all implementation details.

trap door
A hidden software or hardware mechanism that can be triggered to permit system protection mechanisms to be circumvented. It is activated in some innocent-appearing manner; for example, a special ”random” key sequence at a monitor. Software developers often introduce trap doors in their code to enable them to reenter the system and perform certain functions. Synonymous with back door.

Trojan Horse A computer program with an apparently or actually useful function that contains additional (hidden) functions that surreptitiously exploit the legitimate authorizations of the invoking process to the detriment of security. For example, making a “blind copy” of a sensitive file for the creator of the Trojan Horse.

trusted computer system

A system that employs sufficient hardware and software assurance measures to allow its use for simultaneous processing of a range of sensitive or classified information. A system is trusted when it is believed that it can enforce a particular security policy. A CMW level of trust will provide the user and administrator of a system with a given level of trust in its ability to protect data from disclosure.

Trusted Computing Base

The totality of protection mechanisms within a computer system, including hardware, firmware, and software, the combination of which is responsible for enforcing a security policy. A TCB consists of one or more components that together enforce a unified security policy over a product or system. The ability of a TCB to enforce correctly a unified security policy depends solely on the mechanisms within the TCB and on the correct input by system administrative personnel of parameters (for example, a user's clearance level) related to the security policy.

Alternatively, this term refers to the set of hardware and software that together enforce the system's security policy. The TCB comprises only those programs and hardware elements that are known to follow security policy and are considered to be secure. This is necessarily a subset of all the programs available with Trusted IRIX/CMW.

trusted distribution

A trusted method for distributing the TCB hardware, software, and firmware components, both originals and updates, that protects the TCB from modification during distribution and detects any changes to the TCB that may occur.

Trusted IRIX/B

The trademarked name for the trusted operating system that preceded Trusted IRIX/CMW

Trusted IRIX/CMW

The trademarked name (Trusted IRIX/CMW) for the trusted version of IRIX at the B1/CMW level.

trusted path
A mechanism by which a person at a system can communicate directly with the TCB. This mechanism can only be activated by the person or the TCB and cannot be imitated by untrusted software.

trusted process
A process whose incorrect or malicious execution is capable of violating system security policy.

trusted software
The software portion of the TCB.

untrusted process
A process that has not been evaluated or examined for adherence to the security policy. It may include incorrect or malicious code that attempts to circumvent the security mechanisms.

user
Person or process accessing the system either by direct connections (that is,via the system console), or indirect connections (that is, prepare input data or receive output that is not reviewed for content or classification by a responsible individual).

user ID
A unique symbol or character string that is used by a system to identify a specific user.

user profile
Patterns of a user's activity that can be used to detect changes in normal routines.

virus
A self-propagating Trojan horse, composed of a mission component, a trigger component, and a self-propagating component.

vulnerability
A weakness in system security procedures, system design, implementation, internal controls, and so on, that could be exploited to violate system security policy.

vulnerability analysis
The systematic examination of systems in order to determine the adequacy of security measures, identify security deficiencies, and provide data from which to predict the effectiveness of proposed security measures.

vulnerability assessment
A measurement of vulnerability that includes the susceptibility of a particular system to a specific attack and the opportunities available to a threat agent to mount that attack.

work factor
An estimate of the effort or time needed by a potential penetrator with specified expertise and resources to overcome a protective measure.

worm A virus program that has a very narrow purpose. A worm is designed to track down and eliminate specific data. Unlike a simple virus, which by its very nature is obviously present, a worm is designed to remain unnoticed in order that it may continue its task unchecked. Because it may follow a serpentine path in its hunt for particular data it has earned the nickname “worm.”

write A fundamental operation that results only in the flow of information from a subject to an object.

write access Permission to write to an object.

Index

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-3300-002.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
  - On the Internet: techpubs@sgi.com
  - For UUCP mail (through any backbone site): *[your_site]*!sgi!techpubs
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 650-932-0801
- To send your comments by **traditional mail**, use this address:

  Technical Publications
  Silicon Graphics, Inc.
  2011 North Shoreline Boulevard, M/S 535
  Mountain View, California 94043-1389