Trusted IRIX®/CMW Security Administrator's Guide

Document Number 007-3299-002

CONTRIBUTORS

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

Trusted IRIX®/CMW Security Administrator's Guide
Document Number 007-3299-002

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 typographical conventions used, and refers you to additional sources of information you might find helpful.

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

If you have a graphics workstation, you may find it convenient to use the System Manager, which is described in the *Personal System Administration Guide*. That guide should be your first resource for administering graphics workstations. Regardless of whether you use the System Manager or the IRIX command-line interface, the results are the same. The System Manager does not create any new files on your system.

If you have a server, the *IRIX Admin* manual set (of which this guide is part) is your primary guide to system administration, because without graphics you cannot use the System Manager. This guide does not describe the System Manager in great detail. Instead, it covers the traditional shell command approach to administering an IRIX operating system.

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, "Planning Your Trusted IRIX/CMW System"
: Provides a comprehensive discussion of the planning necessary to set up a properly functioning Trusted IRIX/CMW system or network of systems.

Chapter 3, "Administering Login Accounts"
: Provides information on the creation, maintenance, and removal of login accounts under Trusted IRIX/CMW.

Chapter 4, "Networking With Trusted IRIX/CMW"
: Describes the tasks and procedures necessary to administer a network of Trusted IRIX/CMW systems.

Chapter 5, "Administering Access Control"
: Provides information on administering both Mandatory and Discretionary Access Control (including Access Control Lists) under Trusted IRIX/CMW.

Chapter 6, "Administering the System Audit Trail"
: Describes the audit records and methods specific to Trusted IRIX/CMW.

Chapter 7, "Administering Identification and Authentication"
: Describes the Identification and Authentication procedures specific to Trusted IRIX/CMW.

Chapter 8, “Trusted IRIX/CMW System Data Files”
: Describes the system files specific to Trusted IRIX/CMW.

Chapter 9, “Administering Printing and Tape Devices”
: Describes the special actions required to use printers, tape drives, and other media with Trusted IRIX/CMW.

Chapter 10, “Maintaining an Evaluated Configuration”
: Provides information on maintaining security precautions at your site.

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) 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 number

[]	(Brackets) Surrounding optional syntax statement arguments
#	IRIX shell prompt for the superuser (*root*)
%	IRIX shell prompt for users other than superuser
>>	Command Monitor prompt

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

How to Use This Guide

The *Trusted IRIX/CMW Security Administration Guide* is written for administrators who are responsible for performing tasks beyond the reasonable scope of ''end users'' on Trusted IRIX/CMW systems. Frequently, people who would consider themselves end users find themselves performing advanced administrative tasks. This book 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. It is hoped that people who considered themselves end users in the past will, by using this book, gain experience and confidence in successfully performing advanced system administration tasks.

Target Audience of This Guide

This guide is intended for administrators who are responsible for one or more systems running the Trusted IRIX/CMW operating system beyond the usual user responsibility for the user's home directory structure and immediate working directories. This guide and its companion administration guides have been written to provide directions for

those who find themselves in the position of maintaining Trusted IRIX/CMW systems for themselves and others and who require more information about IRIX commands and system and network expertise.

Additional Resources

For easy reference, this section describes the guides and resources provided with your system and the specific focus and scope of each:

IRIX Admin Manual Set

This guide is an additional resource to the *IRIX Admin* manual set. This guide differs from the *IRIX Admin* documentation in certain areas, and this guide should be considered the authoritative guide for the Trusted IRIX/CMW operating system.

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 in the position of 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. They are also available on the World Wide Web at http://techpubs.sgi.com/library. 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* 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* refers to the cc(1) 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 release-specific information about a product. Exceptions to the information in the administration guides are found in this document. Release notes are available online through the *relnotes* command. Each 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 system comes with an online help system. This system 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.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 administration guide has been designed to introduce you to working with secure systems, and in particular with the Silicon Graphics Trusted IRIX/CMW system. This guide gives you recommendations 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 the basic concepts, terms, and features of a trusted system, and explains security procedures and mechanisms. It includes the following sections:

Trusted IRIX/CMW Product Overview

This section 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 see the trusted features of the operating system only 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.

Capability
: 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 Authentication (I&A)
: 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 mechanism 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 mechanism 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 the Mandatory Access Control mechanism that allows the system administrator to restrict access to files, directories, and programs according to security clearance requirements.

Privilege
: 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.

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.

The NCSC has set out evaluation criteria for trusted systems. Trusted IRIX/CMW supports all of the features required for a B3 rating and the assurance requirements for a B1 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 the average or even novice user.

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. Trusted IRIX/CMW will be evaluated at the assurance level as a B1 system. The Orange Book is a common name for the 5200.28-STD Department of Defense Trusted Computer Systems Evaluation Criteria.

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 and parcel 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 in the U.S.A. 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 regularly.

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 (DAC). Trusted IRIX/CMW includes both MAC and DAC, which work together to precisely control system access.

Under Trusted IRIX/CMW, MAC 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.

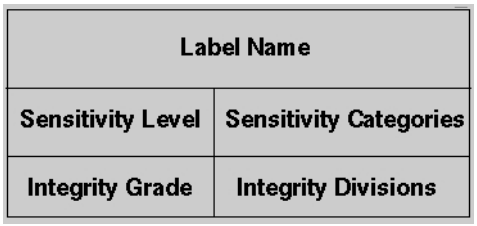

Figure 1-1 Basic Trusted IRIX/CMW Security Label Structure

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 divided into *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 passwords and user clearance data are under mandatory access controls that prohibit access by nonadministrators.

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.

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.

The system administrator implements the level of networking support available at the 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. Each site implementation will be unique.

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 the *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

Planning Your Trusted IRIX/CMW System

Before installing the Trusted IRIX/CMW system, spend some time defining your needs and examining and classifying the various types of information and applications that will reside on your system. Although it is always possible to reconfigure and change your system resources and practices, your installation will benefit from planning. A "dry run" of your trusted system is often beneficial before classified data and users are allowed access.

This chapter describes the planning necessary to ensure a trouble-free and efficient trusted system implementation. If you do not plan your system, you are likely to forget to implement some necessary security policies or labels. While the system is fully configurable at any time, it is inconvenient to have to search through your system and fix existing files and user accounts. Time spent in planning will be much shorter than time spent in repairing and updating.

The following sections are included in this chapter:

Planning Your Security Administration

Administering a Trusted IRIX/CMW system involves many tasks that are performed by the system administrator or divided between the system administrator and the auditor. On a Trusted IRIX/CMW system, these tasks include:

- Planning your security administration
- Defining and maintaining security policies
- Configuring Trusted IRIX/CMW
- Auditing security events
- Maintaining Trusted IRIX/CMW data files
- Controlling access of accounts and groups
- Maintaining password functionality
- Supporting networking with the Trusted IRIX/CMW system
- Maintaining an evaluated configuration

This section explains the privilege environments (augmented superuser and no superuser) in which the administrator works and the privilege mechanisms (superuser-based and capability-based) used, and provides information on the administrator and auditor logins plus tasks they perform on a trusted system.

Privilege Environments

The IRIX system allows many users to work on the same system without interfering with each other or obtaining unauthorized access to each other's work. To accomplish this, the IRIX system defines certain restrictions on the actions of users. These restrictions create an environment in which users can confidently do their work without concern about the safety of sensitive data or loss of system availability.

The restrictions that prevent users from interfering with or inappropriately obtaining the work of others, however, must sometimes be overridden by administrators or system software. The ability to override restrictions is called *privilege*. The means by which the system determines, grants, propagates, and controls privilege is called a *privilege mechanism,* and the set of privilege mechanisms configured for a given system is called that system's *privilege environment*.

The IRIX system provides two privilege mechanisms:

Superuser-based privilege mechanism
: The superuser-based privilege mechanism is the traditional UNIX privilege mechanism. In this mechanism, a process with a user ID of 0 (*root*) has unlimited privilege. When that process executes a command, the process retains the root user ID and, therefore, unlimited privilege. No user ID other than root confers privilege within the superuser-based privilege mechanism.

Capability-based privilege mechanism
: The capability-based privilege mechanism is an IRIX refinement. In this mechanism, user IDs do not themselves confer any privilege. Instead, privilege derives from *capabilities*. A *capability* is a discrete unit of privilege that overrides a set of related system restrictions (see the capabilities(4) reference page). There is no overlap between the set of restrictions overridden by one capability and the set of restrictions overridden by another capability.

  The system may assign capabilities to a user when he or she logs in, based on an entry in the user capability database (see the capability(4) reference page). These capabilities are then available for inheritance by authorized commands. When a process executes a command, the process gets a new set of capabilities. These capabilities are computed from the capabilities presented for inheritance by the process, the authorization of the command to inherit those capabilities, and a set of extra capabilities granted directly to the command itself upon execution. For more detailed information on process and file capabilities see the capabilities(4) reference page.

You use these two privilege mechanisms to configure your IRIX system for one of two privilege environments:

Augmented Superuser

The augmented superuser privilege environment is a combination of the superuser-based privilege mechanism and the capability-based privilege mechanism. In this environment, an administrator logged in as root (user ID of 0) has unlimited, universally inherited privilege. Therefore, the administrator must be careful when executing commands to use only known commands and processes that are appropriate to the task at hand. Use of an untrusted command by the root user could grant unlimited privilege to a Trojan Horse program (a program designed to usurp access rights or privilege while performing an otherwise useful and apparently benign function).

Note that while the superuser-based privilege mechanism grants root unlimited privilege, that privilege does not take the form of capabilities. Capability assignment and inheritance is not affected in any way by the user ID of the process.

The augmented superuser privilege environment also employs the capability-based privilege mechanism. This allows nonroot processes to obtain privilege through specific assignment of inheritance of capabilities. The IRIX system does not automatically assign capabilities to users other than *root* (the system administrator) and *auditor* (the auditor). The IRIX system does, however, assign capabilities to certain nonadministrative commands and daemons. This allows fine control of privilege within these commands when they are executed by a nonroot process.

The augmented superuser privilege environment most closely resembles the traditional UNIX privilege environment. It allows administration without specific concern for capabilities and capability inheritance. The advantage is simple administration and administrative flexibility. The disadvantage is the relative ease with which an administrator or system programmer can make a mistake that results in a security failure.

No Superuser

The no-superuser privilege environment contains only the capability-based privilege mechanism. In this environment the system administrator account is still *root* and the auditor account is still *auditor*, but root no longer derives privilege from its user ID. Instead, all privilege derives from capabilities.

The no-superuser privilege environment yields the greatest protection from administrative error and Trojan Horse attack. It does, however, limit privilege to the set of commands authorized for capabilities, which may not include all commands that need privilege at your site. Also, it limits administrative flexibility by constraining administrators to commands that can inherit capabilities.

The advantage of the no-superuser privilege environment is enhanced assurance that privilege will not be abused or usurped. The disadvantage is somewhat increased administrative complexity and decreased administrative flexibility.

System Administrator

The system administrator defines and maintains the security policy on the Trusted IRIX/CMW system. Policy-related system administration tasks include the following:

- Establishing and implementing a site security policy
- Designing a Mandatory Access Control environment that protects sensitive information without impeding legitimate system use
- Creating and assigning security attributes to new user accounts
- Deleting or restricting access to inactive user accounts
- Protecting the integrity of system databases and executable files
- Managing access to user data
- Monitoring and maintaining the password generation and change system
- Managing network security configurations
- Monitoring system activity
- Checking all new programs and files for security violations before they enter the TCB
- Determining whether adding a program to the TCB is worth the risk involved with having a system that no longer matches the evaluated configuration

Maintaining the configuration database files for security purposes involves the following files. (For more information on the Trusted IRIX/CMW system data files see Chapter 8.)

- */etc/aliases*—the configuration file for e-mail groups
- */etc/capability*— the configuration file of user-allowed capabilities
- */etc/clearance*—the configuration file of user clearances
- */etc/exports*—the configuration file for directories exported using NFS
- */etc/fstab*—the configuration file for filesystems mounted both locally and via NFS
- */etc/group*—the configuration file for group name-to-group ID mapping
- */etc/hosts*—the configuration file for system-to-Internet address mapping
- */etc/motd*—the login message file
- */etc/networks*—the configuration file for network to Internet address mapping
- */etc/passwd*—the configuration file for user name-to-user ID mapping
- */etc/rhosts*—the configuration file that holds a list of possible remote session hosts
- */etc/sendmail.cf*—the configuration file for the e-mail environment on this system
- */etc/shadow*— the configuration file of encrypted passwords
- */etc/sys_id*—the file that holds the name of this system

In addition to defining and maintaining policy, the system administrator also performs day-to-day maintenance tasks that do not require special privilege, ensures system availability, and manages the resources and services provided to the user community.

Administrator login

The system administrator is *root*, regardless of the privilege environment on your system. In the no-superuser privilege environment, the system administrator derives privilege from the */etc/capability* file, specifically the capability setting for *root*:

```
root:all=:all+eip
```

This allows a user who can log in as *root* to specify any set of capabilities at login, as follows:

```
Login: root CAP=all+eip
```

Unless the capabilities are specified at login, *root* has no privilege in a no-superuser privilege environment.

In an augmented superuser privilege environment, *root* may also log in and specify capabilities, but this is not necessary because the *root* user ID automatically confers superuser privilege.

If MAC is installed, the system administrator must pay careful attention to MAC labels. The administrator can log in at any user MAC label and several system MAC labels. The reasons for choosing a given MAC label and the risks associated with operation at each MAC label are described as follows.

User MAC Labels

The system administrator usually logs in at a user label to manipulate data at that label without concern about accidentlly creating an incorrectly labeled file. While this reduces the risk of accidental declassification, it does not eliminate this risk unless the system administrator also avoids MAC related capabilities at login. In the augmented superuser privilege environment, the system administrator cannot use capabilities to limit privilege, so the administrator must still be careful.

Private Database Label (dbadmin)

Certain system databases and log files are protected from modification and observation using the *dbadmin* MAC label. To view these databases and log files without accidentally disclosing their contents, the system administrator is encouraged to log in at dbadmin. In the no-superuser privileged environment, the system administrator can further protect against accidental disclosure by avoiding MAC related capabilities. This is not possible in the augmented superuser privilege environment.

Another reason for logging in at *dbadmin* is to modify private databases. Because the label on a private database is designed to protect it from disclosure, using privilege to edit such a database at a different label creates the risk of accidentally compromising security either by relabeling the database or creating an unprotected temporary file. By editing private databases at *dbadmin* the system administrator avoids this risk.

The final major reason for logging in at dbadmin is to review the system audit trail. The audit trail is kept at dbadmin to prevent users from determining when they are being audited. To avoid breaching this protection, the system administrator should always review audit data at *dbadmin*. This way any files created while reviewing audit data are protected like the audit trail itself.

Except for the ever-present risk of corrupting system databases, there are few risks specifically presented by using the *dbadmin* MAC label.

Public Database Label (dblow)

Databases like */etc/passwd* and most commands are meant to be read or executed by any user, but modified only by the system administrator. To allow global read access while protecting against modification, these databases and commands have the *dblow* MAC label. The system administrator should log in at the *dblo*w MAC label when editing these databases. This ensures that users retain access to the databases and that the databases are never assigned a user label.

Unlike with the *dbadmin* label, there are significant risks to sensitive data when the system administrator logs in at *dblow*. If superuser privilege or MAC related capabilities are present at this label, the system administrator may accidentally copy data from *dbadmin* or a user label into a file at *dblow*. Such a file is universally readable.

If possible, the system administrator should avoid MAC related capabilities when logged in at *dblow*. This is not possible in the augmented superuser privilege environment, and it may not always be reasonable in the no-superuser privilege environment. When superuser privilege or MAC related capabilities are present, the system administrator must take special care to avoid declassifying data.

High Clearance (msenhigh/mintequal)

To import unlabeled data into the system and assign it a label, the system administrator logs in at the High Clearance label. This label allows the administrator to read data at any MAC label regardless of integrity. Data stored at this MAC label are unreadable by nonadministrators. The combination of these two factors allows an administrator at this label to read unlabeled data, store it temporarily at High Clearance, examine it to determine its correct MAC label, and then assign that label without fear of accidental disclosure.

While this is a useful label, the system administrator should always remember that this label confers the authority to ignore the mandatory integrity (MINT) policy of the Trusted IRIX/CMW system. This authority carries the risk of accidentally executing a Trojan Horse program. Also, the administrator should be aware that this label allows private databases to be modified without privilege and, potentially, be created with a label that does not carry MINT protections.

The system administrator is strongly cautioned against functioning at this label. It should be used only for importing unlabeled data.

Auditor

The auditor monitors system activity looking for suspicious events and reports any suspicions to the system administrator for further analysis and possible corrective action. To perform this function correctly, the auditor requires access to the system audit trail and sufficient privilege to control the audit mechanism. The auditor does not need access to user data.

The auditor logs in as *auditor* in the IRIX system. This account has the following attributes, listed in the */etc/capability* file:

```
Clearance (MAC Label): dbadmin
Capabilities:CAP_AUDIT_WRITE,CAP_AUDIT_CONTROL,CAP_KILL
```

The *dbadmin* MAC label, along with DAC permissions on the audit log files, allows the auditor to read the audit trail without privilege. Furthermore, the *dbadmin* label restricts the auditor to reading files at *dbadmin* and *dblow*. User files cannot be accessed because of their mandatory integrity (MINT) labeling. Because the auditor can log in only at this label and has no MAC related capabilities, the auditor is constrained to performing only the auditing function.

Auditing tasks include the following:

- Working with the system administrator to determine the basic set of activities that need to be monitored
- Establishing an auditing environment that effectively monitors these activities
- Reviewing and analyzing the audit logs
- Adjusting the audit environment as needed to obtain specific information to clarify possible suspicious activity
- Reporting possible security breaches to the system administrator
- Working with the system administrator to assess damage resulting from a security breach and design site policy changes to prevent future breaches
- Processing audit data for long-term storage as dictated by the site security policy

Determining the basic set of monitored activities can be difficult. The auditor must balance the desire for detailed information against constraints of storage space and the limited human ability to analyze and assimilate the implications of potentially vast amounts of data. Aggressive auditing can produce large amounts of data about benign user activities. Insufficient auditing, however, may cause the auditor to miss crucial details that are the key to discovering or analyzing an attempt to abuse access to the system.

Some types of activities to be flagged for auditing are as follows:

- attempts at unauthorized entry
- system usage at unusual hours or from unusual locations
- attempts at access control violations
- unexpected use of privilege
- connections with systems outside of the local network
- any activity by particularly interesting subjects
- any accesses of particularly interesting objects
- modifications of system data files
- manipulation of the audit trail itself

Once the system is set up, the auditor is responsible for storing the audit files and periodically viewing and editing the files to produce a record of the system usage. A number of tools are available for use with the system audit trail. Each tool is fully described in its own reference page. Among these are the following:

- sat_select(1M)
- sat_echo(1M)
- sat_interpret(1M)
- sat_reduce(1M)
- sat_summarize(1M)

For a complete discussion of auditing under the IRIX system, please refer to the guide titled *IRIX Admin: Backup, Security, and Accounting*.

Creating Security Policies

You must establish an overall security policy for your site, called a Site Security Policy. A security policy is a series of official statements regarding the rules for the use of the system. The purpose of the policy is to create a clear code to ensure the safe use of the system. This policy should be clearly articulated in a written document available to all users. Each person involved with the secure system should know his or her own security guidelines and responsibilities. At some sites, the security policy itself may be classified.

In 1987, the U.S. Computer Security Act mandated that secure computing sites are responsible for the integrity and confidentiality of their resources and information. Computer systems with sensitive information have long been a favorite attack point for malicious and mischievous intruders. Because of the nature of computers, intruders can do much of their work without direct contact with a human being. Damage done to computers and the information they store can come in many forms. Sometimes the damage is severe, when irreplaceable information is lost or corrupted. Sometimes the damage is a case of theft, when a business competitor seeks advance knowledge of product development. Occasionally, users themselves damage the system, either accidentally or with malice.

A total security policy has a number of segments. Among these are the physical security policy, the procedural security policy, and the system security policy. Take care to ensure that your security policies are concordant and similar in approach to one another. If possible, for consistency, the same person or group should draft all security policies. Avoid complexity in your policies; it can cause users to become confused or to circumvent policy.

Physical Security Policy

A physical security policy is simply the security measures that protect the computer hardware from damage or unauthorized access. Damage in this sense can come from intruders or from aspects of the location, such as water damage or electrical power fluctuations. All physical components of your system, such as the central processing unit (CPU), any storage media, wiring, and remote terminals need to be governed by the physical security policy. Guidelines for effective physical security are:

- Set the PROM password on your system. Instructions for this can be found in your *Owner's Guide* or in the *IRIX Admin: System Configuration and Operation* guide.
- Keep the system physically secure at all times, such as in a locked or guarded room.

- Restrict access to the room to those with immediate need to use the system.
- Request that all users clear the video screen upon finishing work.
- Maintain reasonable security against unauthorized and unrecorded entrance to the entire building or site where a trusted system is used. Such security can be in the form of keyed entry or other clear identification for authorized users.
- Shield and protect all wiring and cables, especially network connections and terminal lines. The lines should be physically covered and unavailable. In no case should any part of the wiring be connected to unsecure systems or any area outside the secure site.
- Keep physically secure all archive media and other data stored on magnetic media. Store it in a locked or guarded media library room. Segregate all media according to the classification of the information contained.
- Restrict access to the media library or libraries to the system administrator.
- Remove, erase, or destroy obsolete data as soon as possible.
- Shred or otherwise destroy paper output when it is no longer needed.

The most secure software is of no use if your physical hardware is vulnerable. Therefore, when you are planning your system, take note of the location of the hardware. The computer itself should be located behind locked doors, and the number of people with access to the room should be strictly limited. Only users, system administrators, and other responsible people should be allowed in the room at any time. This security should never be relaxed. The power source for the computer should always be protected so that the computer is not subject to power surges or momentary power outages. The location of the computer should have a limited number of windows or be totally enclosed.

Beyond this, any coaxial cable connections to other computers on the secure system should be within the same restricted area. If it is necessary to connect to another computer outside the restricted area, the coaxial cable should be routed in such a manner as to avoid convenient points where the cable may be tapped by an intruder.

Peripheral hardware you attach to your system must likewise be protected from intruders. If you use a storage media device such as a cartridge tape drive, you must store the cartridge tapes after use with as much attention, if not more, to security as any other portion of the system. If a cartridge tape falls into the hands of an intruder, it is a simple matter to extract all the information onto a different system. Printers must also be closely monitored, because once information has been printed, it is available to anyone who can read it. Information import and export devices are the weakest link in your trusted system.

System (PROM) Passwords

Silicon Graphics workstations and servers support system passwords. These are also often called PROM passwords, because the firmware that supports the passwording is stored in PROM chips on the system's CPU board. It is strongly recommended that all Trusted IRIX/CMW systems make use of PROM passwords.

These passwords are demanded of the user attempting to access any part of the system while the operating system is not running. For example, with most computers, if you press the hardware reset button, the computer offers you a chance to perform system maintenance before the system reboots. Or in some cases, if you load an installation tape or floppy disk, the system boots from that media instead of booting the usual operating system. Once this has been accomplished, it is a trivial matter for the intruder to mount the disk or disks and gain access to all your files. Because the system password is required before the system does anything but boot the usual operating system, overall security is increased.

The methods for setting the PROM password differ from system to system. To set the system password, see your system's Owner's Guide, or the *IRIX Admin: System Configuration and Operation* guide. The owner's guide can also instruct you on procedures to follow to remove the password if you forget it.

Procedural Security Policy

The procedural policy is the segment of the security policy that dictates how the system is used. It should cover the responsibilities of each administrative user, such as the system administrator (`root` account) and auditor (`auditor` account), and the responsibilities of ordinary users. Guidelines for procedural policy should list decisions concerning these issues:

- Who may use the trusted system and what operations each person is allowed to perform. Keep the number of users as small as possible.
- What system information is available, and through what mechanisms it is available. For example, it is wise to limit the amount of sensitive data that may be printed on paper.
- How information is to be handled at all times.

- How to dispose of old information. Bear in mind that technology exists to read "erased" information from magnetic media. All physical copies of sensitive information should be destroyed rather than simply erased.
- How the system-provided security features are used. For example, you should decide how thorough the auditing of user activities must be, and how often the auditor reviews the audit logs. The responsibilities of the auditor are described later in this chapter.
- How often the Mandatory Sensitivity clearances and classifications are to be reviewed. Also, choose the same schedule for review of the Mandatory Integrity divisions and grades.
- How often and at what level system backups are to occur.
- A procedure and schedule for storage, archiving, retrieval, and disposal of system data.
- A procedure for retiring user accounts for discontinued users.

Procedures used at your site for day-to-day activities can make or break your system security. If procedures are followed, your system security policy is much more likely to succeed than if procedures are lax.

System Security Policy

The system security policy is closely related to both the physical security policy and the procedural security policy. It draws on both the physical and procedural policies to create a total policy for the system. Some guidelines for an effective system security policy are as follows:

- Uniquely identify all users. In no instance should a user be allowed to share an account or any identification with any other person.
- Authenticate each login attempt with a password. Each user account must have a unique password. This password is required for every login.
- Change each user's password frequently. Facilities exist within Trusted IRIX/CMW to generate sound passwords that are difficult for a potential intruder to guess or discover through systematic trial and error.

- Authorize each user to perform only those tasks and to view only the information that he or she needs to complete his or her work, and no more. By reducing the scope of each user account, you reduce the possibility of general damage by an intruder who gains access to a single account.
- Adequately audit each user during every login session. Determine the amount of audit recording necessary to ensure a reasonable knowledge of system activity at all times. Events such as login time; logout time; the creation, modification, and deletion of files; and the invocation of TCB programs should always be audited. If possible, avoid simply auditing everything, because it can result in unnecessary information.
- Educate your users, system administrators, and those who are present at your site but who are not users of the trusted system. Everyone, not only the managers and system administrators, can take responsibility for security.
- Publish the security policies if possible, and make everyone at the site aware of security issues.
- Review the security policies regularly and make necessary changes. As your system works and changes, modifications to the policy may become necessary. A breach of security may necessitate tighter controls and changes to the policy. However, excessively tight controls may deny access and encourage misuse of the system.
- If possible, publish each change to the security policy clearly to all persons at the secure site.
- To ensure compliance, make certain that all levels of management understand and approve each policy.
- Keep the security policy consistent with the goals and standards of your company.
- Do not make the security policies more rigid than necessary. A policy that is unrealistic is likely to be ignored or circumvented by unhappy users.

System security provides for the entire trusted system. The trusted system includes not only the hardware and software but each individual and the group as a whole. All of these combine to form a secure system for the safe development and storage of your sensitive information.

Planning for Users

During preinstallation planning, determine how many user accounts will be necessary for the people who will work on your system. You need to know how many users to expect and how much disk and memory space they require in order to make informed decision regarding hardware resources.

Plan each user's clearance and the specific categories for which the user will be approved. The following list shows some options for setting clearances:

- A user may be granted a single clearance such as *Secret*.
- A user may be cleared for a set of labels such as *Secret*,*A* and *Secret*,*B*.
- A user may be cleared for a range of labels such as *Secret* through *Secret*,*A*,*B*.
- A user may be cleared for a set of ranges of labels such as *Secret*,*A* through *Secret*,*A*,*B* and *Secret*,*B* through *Secret*,*A*,*B*.

For more information on security label structure, see Chapter 5, "Administering Access Control."

Similarly, plan the capabilities required to access key system files, and determine which users can access and manipulate those files. The users with these capabilities are those who have been selected to perform the system administration tasks described in "Planning Your Security Administration." Plan Access Control Lists in working and home directories at this time as well. Information on Access Control Lists can be found in the section titled "Access Control Lists" in Chapter 5.

Examine your hardware at this time to determine how much disk space the users and the information are likely to need. Planning adequate disk allocation at the beginning saves having to reinstall later if your system needs to grow. Include provisions for system audit requirements in system planning. The audit trail requires disk space. The specific amount of disk space required depends on the level of auditing configured and the amount of system activity. Audit trail requirements are discussed in detail in Chapter 6, "Administering the System Audit Trail."

Planning for Mandatory Sensitivity

Before you install your system, you must define your classification scheme for Mandatory Sensitivity (MSEN). You must decide how many levels of sensitivity you need for adequate protection of your data. Bear in mind that some classifications are defined for you and cannot be changed. Among these are *msenlow*, which always represents the lowest sensitivity level on your system; *msenhigh*, which always defines the highest level of sensitivity on your system; and *msenadmin*, for MSEN administration files only. The predefined labels and files are discussed in more detail later in this chapter and in Chapter 5, "Administering Access Control," in this guide. Assigning security classifications and clearances is the responsibility of the system administrator.

Beyond these labels, the system administrator is free to choose and name the clearances and classifications on your system as needed. Remember, however, to use a system that is readily understandable to your users. Familiar terms, such as *public*, *confidential*, *classified*, *secret*, *top secret*, and so on are very useful. If your system of classifications is too complex, users may find it difficult to perform their work.

At this time, you should also choose the categories of information that your system will recognize. This process should be straightforward, because most secure systems deal with only a limited range of information. Also, because a file or resource can be in as many categories as needed, it is easy to make the categories fit the information. Planning the makeup of the files and resources on your system before you begin the installation process is a key step.

Usually, sites converting to Trusted IRIX/CMW move a full complement of existing files to the new system. Each file should be examined to ensure the ongoing security of the system, once the files are installed. You must be certain that no threats to your system are hidden in the bank of files with which you begin.

Planning for Mandatory Integrity

Mandatory Integrity (MINT) works in concert with MSEN to ensure that each user accesses only the parts of the system that he or she specifically needs. While MSEN addresses which files and programs a user may see and invoke, MINT assures that the programs and files are free from threats and trusted by the system for that user. When you plan the clearances and classifications for MSEN, the system administrator should also plan the divisions and grades for MINT. Not all sites find it necessary to use the MINT facilities. MINT can be used to protect your system, but it is not required for a trusted system by the TCSEC (Trusted Computer Systems Evaluation Criteria).

The MINT grades indicate the integrity level of a program or file. Specifically, the MINT grade indicates the quality of the information. Although MINT is a completely separate system from MSEN, the two operate on similar principles. MINT divisions are used to segregate programs by topic, similar to an MSEN category. For example, system administrators can have a MINT division for their utilities and tools, and software developers can have another MINT division for their programs and tools.

Each user has a MINT grade and zero or more divisions in their label. A file or program created by that user automatically inherits that user's MINT grade and divisions (if any). If a user's MINT label has no divisions, then that user's MINT grade must be met by any program that the user accesses. If a user's MINT label contains divisions, a user is constrained to access only programs and files within those divisions. Make certain that each user has appropriate access to files and resources, and plan your MINT system appropriately. Use MINT to keep your sensitive accounts safe from invoking suspect programs that may cause damage.

The primary function of MINT is to make it possible to limit access to programs based on the known quality of the program. MINT protects powerful user accounts, such as root and auditor, from being corrupted or deceived by dummy files of low integrity. A secondary MINT function is its use in tailoring the tool set available to each account so that every user has an appropriate tool set and does not have access to inappropriate tools.

Clearly, the known integrity of a user and the task of the user should be used as guidelines in assigning the integrity level of the user account. Because a file created by that user inherits the user's integrity level and will thereafter be available to other users of a similar label, it is important to trust each user at the assigned integrity level. MINT makes it possible to disallow high-grade users from viewing files and invoking programs created by less trusted users.

Planning for Auditing

While you are planning your system, plan to use the system audit trail features effectively. Decide what kinds of events you wish to audit and how much information you wish to store. If disk space is limited, you may wish to restrict auditing to events such as file removal, attempts to access the system, and denial of service. You must also budget the time of the auditor (or system administrator) to review and reduce the audit log to evaluate the security status of your system. See Chapter 6, "Administering the System Audit Trail," for more information about auditing.

Planning for Networking

If you plan to have more than one Trusted IRIX/CMW system at your site and you would like to connect them as one logical system, if you have a single Trusted IRIX/CMW system you would like to connect to your existing network, or if you have a totally heterogeneous network, you should plan a networking strategy. First, become familiar with standard IRIX networking software. Then, after reading Chapter 4, "Networking With Trusted IRIX/CMW," map out your proposed network and plan your installations accordingly.

Configuration Files

The following is a list of important items to keep in mind as you install and use your Trusted IRIX/CMW system:

- The label of */etc/group* is *dblow*.
- The label of */etc/passwd* is *dblow*.
- The label of */etc/clearance* is *dblow*.
- The label of */etc/shadow* is *dblow*.
- The label name *dblow* translates to *msenlow/minthigh*.
- The label name *dbadmin* translates to *msenadmin/minthigh*.
- The integrity of all administrative databases is *minthigh*.
- The sensitivity of administrative databases is either *msenlow* (publicly readable) or *msenadmin* (hidden from users).

Identifying the System

If you are unsure whether Trusted IRIX/CMW has been installed on a system, use the *sysconf* command to list MAC, ACL, or capabilities status. The following example shows a *sysconf* query and the system response that MAC is installed:

```
sysconf mac
mac=1
```

The following *uname* command returns the name of the operating system currently in use on the system:

```
uname -s
```

Similarly, the *uname* system call returns the operating system name from within a program. Such a test can be useful when writing code that may or may not be run on a trusted system.

Installation Notes

Installation of Trusted IRIX/CMW is substantially similar to that of standard IRIX. Installation instructions are found in the *IRIX Admin: Software Installation and Licensing* guide or in the release notes and errata.

Keeping Your System Installation Trusted

Trusted IRIX/CMW is part of the Trusted Computing Base (TCB), that is, the hardware and software that have been determined to be trustworthy. Making any changes, including deletions, to the TCB results in a system that is less trustworthy than the original. If your site has been determined to require the evaluated configuration of Trusted IRIX/CMW, it is vital that nothing in the TCB be changed or removed, and that no additional hardware or software be added to the TCB.

Regenerating the TCB

If, at some point, you discover that new software has been added to your TCB, your system can no longer be assured to be trustworthy, and the evaluated configuration will have been altered. This does not mean that there has been a security breach or loss or disclosure of data, merely that the safeguards built into Trusted IRIX/CMW have been weakened.

You can get back to the evaluated configuration by storing your data files on tape, removing the operating system, erasing the disks completely (reformatting the disks will generally suffice, unless you have specific requirements for reusing magnetic media), and reinstalling the operating system *from the original distribution media*. It is vitally important that you go back to the original distribution software to be sure that you are installing the evaluated configuration.

Once this is done, reinstall your configuration files and system environment, with the exception of the file or files that were added to, changed in, or removed from your TCB. Your TCB should then again be reliable.

System Administration Tools

The standard IRIX system administration tools are used to manage Trusted IRIX/CMW. For complete information on using these tools, consult your standard IRIX documentation.

Maintaining Administrative Files Under RCS

When making administrative changes to files under the Revision Control System (RCS), you can use a standard editing utility, such as *vi*, to be sure the correct audit events are generated. The editor (user account) privilege must match that of the file for editing to be allowed.

User Information Files

The following files hold information about your user accounts:

- */etc/clearance*
- */etc/shadow*
- */etc/passwd*
- */etc/group*
- */etc/config/login.option*

MAC Label Text Description File

The following file controls MAC labels on your system (labelnames, levelnames, gradenames, categorynames, divisionnames):

- */etc/mac*

Hostname and Network Services Files

The following files hold information about hosts on your network and about your system:

- */etc/hosts*
- */etc/networks*
- */etc/hosts.equiv*
- */etc/inetd.conf*
- */etc/resolv.conf*
- */etc/services*
- */etc/sys_id*
- */etc/net_id*
- */etc/sys_owner*
- */etc/config/ifconfig-1.option*s
- */etc/config/ifconfig-2.options*
- */etc/config/ifconfig-3.options*
- */etc/config/iflabel-1.options*
- */etc/config/iflabel-2.options*
- */etc/config/iflabel-3.options*
- */etc/config/netif.options*

Filesystem Files

The following files contain important information about your filesystems:

- */etc/config/automount.options*
- */etc/fstab*
- */etc/exports*
- */etc/lvtab*

Mail Aliases Filest

The following files contain information used to send mail to users on your system:

- */etc/aliases*
- */etc/sendmail.cf*
- */etc/config/sendmail.mac_label*

User Messages Files

The following files hold messages for your users:

- */etc/motd*
- */etc/issue*

Terminal Configuration Files

The following files configure the ports through which users can log in and to which modems and other I/O devices may be connected:

- */etc/gettydefs*
- */etc/ttytype*

Auditing Files

The following files control your system auditing environment:

- */etc/config/sat_select.options*
- */etc/config/satd.options*

System Clock Files

The following files pertain to your system clock:

- */etc/TIMEZONE*
- */etc/config/timed.options*
- */etc/config/timeslave.options*

Deactivating a Trusted System

When the time comes to deactivate a particular computer from your system, or perhaps deactivate your entire site, you must take certain steps to ensure that information is not inadvertently disclosed:

- All magnetic media not being maintained in a secure manner must be thoroughly erased or destroyed. Except for project records and results that must be maintained, destroy all backup media and other records that might be salvaged if they were merely discarded.
- All accounts on all systems must be thoroughly deactivated.
- All disks on all systems must be reformatted or destroyed.

Chapter 3

Administering Login Accounts

Two classes of login accounts are found under Trusted IRIX/CMW: user accounts and administrative accounts. The administrative accounts are *root* (the system administrator) and *auditor* (the auditor). All other accounts are ordinary user accounts. Administrative accounts are discussed in the section "Planning Your System Administration" in Chapter 2. This chapter discusses the appropriate use and management of user accounts.

User accounts are both the first line of defense of a trusted system and potentially the weakest link in that system. Every user account can break system security if it is not managed well, and every user account can be used to enforce system security. The way your user accounts are managed is crucial to a successful secure system.

Users must have ready access to the files and resources they need to perform their work. If this access is not available or is inconvenient, users circumvent the security policies and create threats to system security. However, users should also not be allowed access to unnecessary files and resources, because this is a security threat in itself.

Guidelines for effective secure management of user accounts are explained in this chapter. Procedures for administering user accounts and user groups are also presented.

This chapter includes the following sections:

User Accounts

The following sections give guidelines and instructions for creating user accounts:

- Guidelines for User Accounts
- Creating User Accounts
- Removing a User
- Changing Clearance Information

Guidelines for User Accounts

Guidelines for user accounts are listed below:

- Always use a different account name and user ID number for each user on your system. Each account should represent only one person, for accountability.
- Always create passwords for all accounts on your system.
- Never assign a login name that begins with a number. Some networks do not interpret these login names correctly.
- Always choose unique user identification names for your users. For example, the login name *steveb* is a better choice than *user001*. A login name and the other information associated with an account should always be readily associated with the person who owns that account. It is generally possible to find distinguishing personal characteristics to differentiate between two or more users with similar names.
- Include the user's full name and some personal identification, such as job title and phone number, in the comment field of the */etc/passwd* file. Be careful, however, not to include classified information in the */etc/passwd* file.
- When you create user accounts, be certain that the user's environment is properly initialized for security. For example, in the *.profile* or *.cshrc* files, set the user's UMASK to 077. This initializes the default DAC permissions to allow the user to access only those files he or she creates.
- In the *.profile* or *.cshrc* files, set the PATH environment variable to include only those directories that the user is allowedto access. Also, in the PATH variable, make certain that the user's home directory is searched last, after the system directories, for commands. This guards against some forms of Trojan Horse attack. Do not include any temporary or public directories in the PATH, such as */tmp*.

- If possible, place a copy of the security policy in each account.
- When you remove a user account, first make a backup tape of all files in the home directory belonging to that account.
- When you remove a user account, assign new owners to any files on the system still owned by the removed user.

Creating User Accounts

This section gives directions on creating user accounts. Before beginning the process, choose the user's login name, user ID number, allowable security labels (or label ranges), and any administrative roles.

On a trusted system, shadow passwords (*/etc/shadow*) are always used; see the *pwconv*(1) reference page. When MAC is installed, every user must have an entry in */etc/clearance*. All of these databases, except */etc/passwd*, are protected from perusal by nonprivileged users.

It is important to follow the procedures exactly as they are specified in this guide. These procedures often involve manipulating sensitive system access files. Failure to follow the exact procedures listed here could leave your system without the designed security protections.

Administrative accounts

The Trusted IRIX/CMW system has an administrative account (*root*) and auditor account (*auditor*). Do not change these accounts.

All nonsecurity-related system administration should be done using the *cadmin* tools. These tools verify that the invoking user is allowed to perform the necessary role.

Creating Normal User Accounts

New user accounts must be created by the system administrator. Adding auser requires these steps:

1. Create a *passwd* file entry and a home directory for the user.
2. Create entries in the */etc/shadow*, */etc/clearance*, and */etc/capability* files for the user and use *chlabel* to set the MAC label on the user's home directory.

To create the */etc/passwd* file entry, perform these steps:

1. Enter the command:

   ```
   vi /etc/passwd
   ```

2. Add a line with the new user's information of this form:

   ```
   username:*:UID:GID:Comments:home-dir:default-shell
   ```

   Here is an example line:

   ```
   jeffz:*:1998:50:Sample User:/usr/people/jeffz:/bin/csh
   ```

To create the user's home directory, perform these steps:

1. Change directories with the *cd* command to */usr/people*.
2. Use the *mkdir* command to make a home directory for your new user. Give the directory the same name as the new user.
3. Use the *chown* and *chgrp* commands to change the ownership of the new directory to the new user.
4. If your new user is using the C-shell (*/bin/csh*), copy the files */etc/stdcshrc* and */etc/stdlogin* to the new directory, naming them *.cshrc* and *.login*.

To edit the */etc/clearance*, */etc/capability*, and */etc/shadow* files, perform these steps:

1. Add an entry to the */etc/clearance* file according to the format described below. The example shows */etc/clearance* entries for an auditor (*audry*), two system administrators (*andy* and *amy*), and an operator (*oswald*). Andy is also allowed to be an operator. All of these users do "real" work on the system except for *andy*, who is a full-time system administrator. Everyone but *andy* is cleared for *userlow*, which is their default label:

   ```
   audry:userlow:dbadmin userlow
   amy:userlow:dblow userlow
   andy:dblow:dblow msenhigh/mintlow userlow
   oswald:userlow:msenhigh/mintlow userlow
   ```

2. Add an entry to the */etc/capability* file according to the format described below. The following are */etc/capability* entries for a number of users. Note that the *dbadmin* account has a master capability that includes all defined capabilities:

   ```
   auditor:CAP_AUDIT_WRITE,CAP_AUDIT_CONTROL,CAP_KILL+eip
   dbadmin:all=:all=
   ernie:all=:CAP_FOWNER,CAP_SETFCAP+eip
   bert:CAP_FOWNER,CAP_ENGR,
   ```

   An */etc/capability* file entry includes the account name and the capabilities associated with that account in a comma-separated list.

3. Add an entry to the */etc/shadow* file according to the format described below. The following are */etc/shadow* entries for two users:

   ```
   root:kEXFeXFTPoxE
   bill:6k/7KCFRPNVXg,z/
   ```

   An */etc/shadow* file entry includes the user's account name and encrypted password, separated by a colon (:). When you add an entry, you need only add the account name and a colon; the *passwd* utility encrypts and enters the password.

When changing the labels for the new user, follow these steps:

1. Use the *chlabel* command to change the label of the files in the directory to the lowest allowable label of the new user. You must use the lowest allowable label so the user can access those files without regard to his or her login label.
2. Use the *chlabel* command to change the label of the new directory to the lowest allowable label of the new user. You must use the lowest label or the user cannot find his or her home directory when logging in at the lowest label.

Additionally, the system administrator should set a password for the new user, using the following commands:

```
passwd jeffz
```

and

```
passwd -f jeffz
```

The first command creates a password for *jeffz*. This password must be selected by the system administrator and told to the new user. The second command forces the new user to change the password at the first login.

Removing a User

When a user has finished all use of a secure system, that user's account should be closed quickly. It is the system administrator's concern that unauthorized users not be allowed on the system, and he or she needs to be informed at once when a user leaves or ceases to use the system. The system administrator should replace each of the following with the string "*INVALID*": the former user's encrypted password field (in */etc/shadow*), both capability lists in */etc/capability*, and both clearance fields in */etc/clearance* . The entries in the files should not be removed. The system administrator should also check for *crontabs*, *at* jobs, or print jobs the former user may have queued.

Once the user is removed, check all system files and change ownership of any files on the system that are owned by the defunct user account. If the user had access to other accounts, change the passwords on those accounts immediately. Also, remove the user's name from all groups on the system.

Changing Clearance Information

The security clearance information assigned to a user may be changed by the system administrator by updating the appropriate entry in the */etc/clearance* file.

If the user's new security clearance includes all of his or her old labels, that user may remain logged on to the system and active while the clearance is updated.

User Groups

On a trusted system, you typically have one or more confidential projects at any given time. Also typically, the users working on those projects need to share files and resources. To accommodate this need, you can create user groups. DAC provides a set of permissions for a file owner's group, as well as for the owner of the file and the whole user community.

Trusted IRIX/CMW provides for multiple concurrent groups. That is, a particular user can be a member of any number of groups, or even of all groups on your system. When you log in, your group ID is set to the group ID in your entry in the *passwd* file. To change to a different group, use the *newgroup* command.

Group your users based on their common needs. Put all the users on a given project in the same group. All members of a group acquire the group ID in addition to their user ID when they log in. Using the DAC permissions and appropriately defined Mandatory Access Control (MAC) permissions, it is possible to give each member of a project team complete access to necessary files and exclude other users from confidential files.

Guidelines for User Groups

Guidelines for user groups are as follows:

- Place users working on the same project or who have similar needs in a group. Consider, for example, a group of data entry clerks. Users with similar needs may work on different projects, but they all need similar tools and resources.
- Add a group at the same time you add each new project to your system.
- Assign each group a unique and readily identifiable group name. For example, *motordev* is a better name than *group001*.
- Never begin a group name with a number, because this can be misinterpreted by the system.
- The file */etc/group* maintains a list of the valid groups and their members. It is possible to edit the */etc/passwd* file and change the ID number of a given group. No checking is done between these two files, and the system administrator must make certain that all user IDs and group IDs given in these files are correct.
- Run the *pwck* program frequently to check your system for potential problems in the */etc/password* file.
- It is sometimes desirable to create a group containing only a single user who is performing specialized work.

Adding a New Group

To add a group, the system administrator logs in at the label *dblow* and performs the following steps:

1. Enter the command:

   ```
   su dbadmin
   ```

2. Enter the command:

   ```
   vi group
   ```

3. Add a line for the new group in this form:

   groupname`:*:`*username*`,`*username*`,`*username*

4. Exit *vi* and the *dbadmin* account.

Removing a Group

When a group has no more users, or a project group has finished all work, the group should be nullified. You should not, however, remove a group entirely, because the possibility exists that the same group name or ID number might be reused, creating a security hazard. To remove a group, edit the *group* file in the same way as to add a group, and remove all usernames from the entry for the defunct group. This way, the group is effectively removed, but the entry remains and so cannot be reused.

At your convenience, search through the system and find files that are owned by the defunct group and change their ownership to another group or remove them.

Chapter 4

Networking With Trusted IRIX/CMW

This chapter is designed to educate the system administrator on the differences between communications under Trusted IRIX/CMW and under standard IRIX. It is assumed that the system administrator is familiar with networking software as described in the guide titled *IRIX Admin: Networking and Mail*. Where information in this chapter differs from the information presented in other IRIX documentation, this chapter must be considered authoritative for Trusted IRIX/CMW.

The following sections are included:

Introduction to network security

A central concept to the Silicon Graphics product family is that the most effective way to bring computing power to the individual is to connect the workstations and produce a unified system of great power. The heart of the Trusted IRIX/CMW multiple workstation system is the network. This connection allows the workstations that make up a Trusted IRIX/CMW system to share a common set of files by use of the NFS filesystem sharing protocol. It also allows computing resources to be shared via TCP/IP protocols. This mechanism allows file resources to be shared among users, who find a single filesystem presented to them, while still providing the convenience and security of a personal workstation.

The key premise of trusted networking is that arbitrary and capricious information is never encountered. In order for this to work, the networks must be physically secure, the systems connected to the network must be appropriately administered, and the systems may either be trusted to enforce the network security policy, or are otherwise trusted and cleared to the network high label at all times.

The network is a critical part of each system's TCB and must be subject to the same security standards that prevent inappropriate access to your Trusted IRIX/CMW workstation. The physical hardware of the Ethernet cable and the connection points between the workstations and the cable must be secured.

Many users assume that any number of separate physical computers connected via coaxial cable defines a network. This, however, is not the case. A network is an uncontrolled data communication system, and not only can arbitrary and capricious information be encountered, but it must be expected.

The network under Trusted IRIX/CMW is a restricted system interconnect. The security of the system depends upon the proper use of the hardware and the software. 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.

For a more general discussion of networking theory and practice, the following books, available at any computer bookstore, are recommended:

Internetworking With TCP/IP—Principles, Protocols, and Architecture, by Douglas Comer, published by Prentice-Hall.

UNIX Network Programming, by W. Richard Stevens, published by Prentice-Hall.

Theory of TSIX Networking

Mandatory Access Control is one of the primary means by which the system enforces your security policy. In order for your network to be trusted, the network must also support the mandatory access control rules of your security policy.

Not all sites may be homogeneous; that is, systems from different vendors may be on your trusted network, and perhaps not all are trusted multilevel systems. Yet all systems need to interoperate. The TSIX standard has been created to expand IP support of MAC labels in a way that allows various vendors and trusted sites to interoperate.

Under the TSIX system, labeling occurs at two levels:

- At the network interface level, IP Security Options are used to route traffic. There are several forms of IP Security Options, of which the most common are RIPSO (Revised Internet Protocol Security Option) and CIPSO (Commercial Internet Protocol Security Option). A received packet either has CIPSO or RIPSO options, or is unlabeled.

  In the case of CIPSO options, the sensitivity label is extracted, and if it is not within the label range of the interface it is dropped. In the case of an unlabeled packet, the sensitivity label is obtained from the default label of the interface if present, or from the host or network entry in the */etc/rhost.conf* database. If the default label is not within the range of the interface, the packet is dropped.

  An integrity label range may be specified for the network interface. If an integrity range is present, the integrity portion of the label from the SGIPSO tag is used for the label range comparison; otherwise, the default integrity for either the interface or host is used similar to the procedure for unlabeled packet processing.

  For packets that are simply routed, or that require a reply from the TCB (such as ICMP echo packets), the outgoing packets receive the same label as the incoming packets. That label is used for a label range check against the outgoing interface, and the packet is dropped if it is not within range.

  For TSIX hosts, the IP header label is no longer used for policy. For unlabeled and non-TSIX hosts the IP label is used for any further policy decisions.

- At the CMW session manager level, Security Attribute Modulation Protocol (SAMP) and Security Attribute Token Mapping Protocol (SATMP) are used to send all the security attributes required to enforce security policy between network components.

  For TSIX hosts, the security attributes are provided in the SAMP header. Attributes identified as mandatory that are not present in the SAMP header are first supplied from the interface and then from the *rhost.conf* entry. The */etc/rhost.conf* file is a database of information about the network security idioms used by all hosts in your trusted network. If not all mandatory attributes are present, the packet is dropped in the case of UDP, or the connection is closed for TCP. The session manager maintains a composite set of attributes for the socket that consists of the last modulated attributes and any defaults. These composite attributes are the attributes used to enforce policy on delivery to applications, and are available to trusted applications through the TSIX API.

  For UDP SAMP, attributes accompany each packet. For TCP, on initial connection the full set of attributes are exchanged before control is passed back to the application. If the attributes received from the remote host are not within the range of the user process, the connection is dropped with *reset*. A server process never sees the failed connection attempt, and the client sees the message `connection closed by remote host.`

TSIX Security Policy

The policies enforced by trusted networking are as follows:

- Received packets must be within the label range of the interface and the host or network.
- Delivered data must have a label dominated by the receiving process.
- Trusted processes may themselves enforce appropriate policy.

Trusted Network Preparation and Configuration

The following sections deal with the steps involved in preparing a number of Trusted IRIX/CMW workstations to operate on a network. The possibilities in a heterogeneous environment are too numerous to document, but enough background is provided to instruct you in the necessary tasks that must be performed.

To create your Trusted IRIX/CMW network, you must first assemble your system and network hardware according to the instructions provided in the hardware package. Once the system is assembled, install your Trusted IRIX/CMW system distribution media on each system and you are ready to begin.

Caution: Your trusted network operates most smoothly if the security labels used on each system are the same, and have the same meanings, as those on every other system (that is, one set of labels for the network). Careful planning before you create your trusted systems is of great value when you create your network.

Note that all network daemons, including *nfsd* and *inetd*, must be run at the same (or equivalent) labels on each system on your network. This is usually the administrative low label, *dblow,* on Trusted IRIX/CMW systems. If this is not the case, the daemons cannot find each other and communicate, and thus do not function properly. Label cognizant daemons such as *nfsd* and *inetd* enforce the MAC policy themselves, instead of relying on the operating system to enforce it for them, but all other network-related daemons (such as *timed*, *routed*, *yp*, and so on) are not label-cognizant and must always be interoperated at a single pre-defined label.

Some sites find that all systems on the network share the same set of labels (such as *unclassified*, *confidential*, *secret*, *top secret*, and so on) and while the relationship between the levels is the same, the integer values to represent them may be different. For example, the integer value of *confidential* is always less than the integer value of *secret*, but one system might have specified the *secret* label with an integer value of 3, and another may have used an integer value of 30 for the same *secret* label. This creates the need for software to interpret and equalize labels if all the systems do not share an identical label set.

Add to these situations the extra levels of specificity used to create MAC categories, and the process of defining the rules of translation for labels from different systems can become quite complex.

Labels transmitted on the network are tagged with a Domain of Interpretation (DOI). These tags tell each system how to map the bit representation of the incoming sensitivity label into the native bit representation.

As long as all systems in your network have the same native representation (they require no mapping), the network MAC policy works transparently. The network uses DOI 3 (which means native representation as determined by the local label database) and all packets with DOIs other than 3 are dropped.

Label Restrictions on Network Services

All network services bear the restriction that the user must be cleared on the remote system at the label he or she is currently using on the local system. For example, if a user is logged in at the label *usermiddle* on a system and attempts to rn the *rlogin* program on a remote system, the attempt cannot succeed unless the user is also cleared for the *usermiddle* label on the remote host. If the user's maximum clearance is *userlow* on the remote system, the *rlogin* attempt fails.

When using network services to log in to remote hosts, no provision is made for entering a label on the remote host. The user must conduct all transactions with the remote host at the current label on the local host. This restriction prevents writedown of data.

Creating an Interoperating Heterogeneous Network

If you have a heterogeneous environment, the following steps should help you create an interoperating network. Trusted systems from different vendors are not likely to have the same feature sets, and those features that are similar in use may be implemented in radically different ways. Follow these steps:

1. Every system should have DOI set to 3.
2. Determine a set of labels and categories to use and their integer values.
3. Change all label databases to match the selected set of network labels. On your Trusted IRIX/CMW systems, edit the */etc/mac* file.
4. Specify a default integrity label for the network. The incoming sensitivity label and the default integrity label are combined on each packet to create the MAC label that is used to enforce your security policy.

For more information on interoperating and trusted systems, read the material provided on the Trusted Systems Interoperability Group (TSIG) site on the World Wide Web (http://www.sterling.com/ftp/tsig).

Creating a Homogeneous Network of Trusted IRIX/CMW Systems

Trusted IRIX/CMW has significant differences from other trusted operating systems. One of these differences is the SGIPSO2 protocol. This protocol includes administrative labels, integrity labels, and user IDs. If you use a full Trusted IRIX/CMW label set, you can only fully make use of your label set with other Trusted IRIX/CMW systems.

The full label set is required by Trusted IRIX/CMW systems to enforce security policy in the evaluated configuration.

Trusted systems generally have a system high label—a label that dominates all other labels on the system, and a system low label—a label that is dominated by all other labels on the system. These labels are generally required for administrative activities. For example, when performing a system backup you must be logged in at the system high label. In trusted systems other than Trusted IRIX/CMW, these labels have to be calculated based on the configuration in the label database.

Trusted IRIX/CMW has implemented label types. There are label types for the high label, the low label, the auditor, and a type for ordinary labels. There is no mechanism except for the SGIPSO2 protocol for communication of administrative labels to nonTrusted IRIX/CMW systems. A similar issue is raised by using default integrity labels. The full range of labels that can be represented on a Trusted IRIX/CMW system cannot be communicated across the network. This is not an insurmountable problem, however, because if your users know what labels are cleared for network use, they can change their operating label to an acceptable label before using the network.

In order for a system to communicate with another host on your network:

1. The host must have an entry in the *rhost.conf* database. (Wildcard entries are permitted.)
2. The “next hop” host must have an entry in the *rhost.conf* database.
3. The interface must be labeled using the *iflabel* command.
4. The packet must have a label within the range of all of the above conditions.

rhost.conf Database

The *rhost.conf* database describes the MAC label interaction between your system and every other system on your trusted network. This database describes default labels to use for networking to non-trusted hosts and the idiom of network label support used for trusted multi-label hosts. All information relevant to trusted networking for each host on your network is stored in this file on each system. You must manually enter each host in your trusted network in your *rhost.conf* file, along with the idiom of label support used by that host.

To create your *rhost.conf* database, take the sample */etc/rhost.conf* file distributed with Trusted IRIX/CMW and edit it, adding each host's entry individually.

rhost.conf File Syntax

The following sections describe the syntax of the *rhost.conf* file:

- Templates in rhost.conf
- Individual Host Entries in rhost.conf
- Wildcard Entries in rhost.conf

Templates in rhost.conf

The information at the top of the *rhost.conf* file specifies templates for hosts that fulfill certain common requirements for label interchange. The syntax allows you to create custom host templates as well. At the bottom of the *rhost.conf* file, an entry that is similar to this

```
Hostname:    default_spec = default_sgipso:
```

is specifying a previously defined template (in this case, *default_sgipso*, which is reprinted in the following template).

If you have planned your trusted network well, using a common set of labels, you should be able to use the provided templates as distributed for your hosts. Several templates are provided for commonly used labeling idioms.

The following template is reprinted here as an example:

```
######################################################################
# SGIPSO2 Trix5.3/4.0.5 Hosts
#
######################################################################
default_sgipso: nlm_type = sgipso2:\
smm_type = single_level:\
doi = cipso_tsig:\
default_spec = .:\
max_privs = CAP_CHOWN,CAP_SETGID,CAP_SETUID,CAP_DAC_EXECUTE+eip:\
cache_size = 512:\
flags = mand_sl,import,export:\
def_privs = CAP_CHOWN,CAP_SETGID,CAP_SETUID,CAP_DAC_EXECUTE+eip:\
def_luid = root:\
def_uid = root:\
def_gid = root:\
def_ngrps = 1:\
def_gids = sys:\
def_sid = 0:\
def_sl = userlow:\
def_clearance = userlow:\
def_ilb = userlow:\
min_sl = userlow:\
max_sl = userlow:
```

Individual Host Entries in rhost.conf

A specific host entry in an *rhost.conf* file follows the same syntax as a template entry, and can either specify all attributes, or use some combination of a template and an individual specification:

```
# anewhost:\
#       default_spec = default_tsix_1_0:\
#       min_sl       = confidential:\
#       max_sl       = secret:
```

This example uses a template to describe most attributes, and the minimum and maximum label are specified for the host.

The localhost must always be specified as well:

```
# Always put host and localhost
host:           default_spec = default_tsig:
localhost:      default_spec = default_tsig:
```

The following examples show how individual hosts may be specified using templates. Note that hosts may be specified by name or by IP address. Also, note that the second entry is a wildcard entry, which is described in "Wildcard Entries in rhost.conf."

```
pender:       default_spec = default_sgipso:

154.16.0.0:   default_spec = default_single_level:

192.88.83.26: default_spec = default_single_level:

howard:       default_spec = default_tsig:

hancock:      default_spec = default_cipso:
```

Wildcard Entries in rhost.conf

A wildcard entry is an IP address with zeros in some positions. For example, consider the following IP format addresses from an *rhost.conf* file:

```
128.01.01.0:
128.01.0.0:
128.0.0.0
0.0.0.0:
```

All these are wildcard entries with increasing scope as you move down the list. When looking up a specific network address in response to a user command, a system must first search for the whole address, and then successively withdraw a byte and look for a wildcard match. This allows the following strategy to work: Consider that a user has requested a file transfer from a specific host. The operating system must determine what label interchange arrangement exists with that host, so it begins by looking up the host's entry in the *rhost.conf* database. The IP address being searched for is 128.01.01.01. The *rhost.conf* database could hold any of the following entries:

128.01.01.01: This is the host that is needed. Use its specific entry from the database.

128.01.01.0: This entire subnetwork is defined to use a single entry from the database (partial wildcard entry).

0.0.0.0: The whole network is untrusted and is communicated with only at the default label (100% wildcard entry).

The rhost Command

The *rhost* command, without any options, loads the default configuration file, */etc/rhost.conf*, into your kernel for use by the network. The following options are available to the *rhost* command:

-l hostname Displays the characteristics of the current host

-f filename Loads an alternate file into the kernel.

The syntax of the input file is modeled after the *rhost.conf* database description found on the TSIG site on the World Wide Web (http://www.sterling.com/ftp/tsig).

iflabel Command

The *iflabel* command is used to assign a specific set of trusted networking rules (called an idiom) to a network interface and to configure the network interface label parameters. This is done because trusted networking requires that all packets sent over the network must retain their security classifications and other attributes, unless specific policy is in place. Because your trusted system may be in heterogeneous operation with other trusted and untrusted systems, several idioms of network labeling are allowed to maximize interoperability.

For complete information on the use and parameters of *iflabel*, see the *iflabel*(1) reference page.

The *iflabel* command is invoked at boot time from */etc/rc2.d/network* to define the label exchange idiom of each network interface present on the system; you may also use it once the system is running to redefine an interface's idiom or other label parameters. However, it is generally considered good system administration practice to change the *iflabel* configuration file and then to reboot the system in order to change the idiom of a network interface. There is one configuration file, called */etc/config/iflabel-?.options* for each network interface on your system. The actual filenames replace the question mark with the code for the particular interface, for example, *ec0*. Use the *netstat -i* command to list your system's network interfaces.

Only the superuser may modify the configuration of a network interface. The *iflabel* command may be invoked with any of several idioms, or with none at all. If *iflabel* is invoked with no idiom or label parameters, it displays the current idiom for the specified network interface and the current values of the parameters appropriate to that idiom.

The *iflabel* command may be invoked with any one of these idioms:

MONO — This idiom is used when a trusted multi-level host (such as a Trusted IRIX/CMW system) is attached to a single level (possibly untrusted) network. Datagrams sent between hosts on a single level network do not contain labels in their IP option subheader; all traffic on a single level network is defined to take place at the label of the network.

BSO — All network traffic over a BSO interface must be at a label dominated by the highest label of the network interface and dominating the lowest label of the interface.

BSO_TX — This idiom is precisely like BSO except that labels are always transmitted with outgoing datagrams but not required on incoming datagrams.

CIPSO2 — All network traffic over a CIPSO2 interface must be at a label dominated by the highest label of the interface and dominating the lowest label of the interface.

SGIPSO2 — This idiom, an extension to CIPSO2, provides a mechanism to attach to all network traffic, a complete Silicon Graphics MAC (mandatory access control) label, along with the user ID of the sending process. Besides a sensitivity level and a category set, the MAC label may include integrity components and label types other than TCSEC, EXACT, and BIBA.

Domains of Translation and Interpretation (DOT and DOI)

You can create an interoperating heterogeneous network with a limited label set, but if you wish to make full use of Trusted IRIX/CMW features, such as the windowing system and shared filesystems, you must use the full set of security attributes to appropriately enforce the security policy of the system.

The TSIX standard is a specification of a session layer protocol for passing all attributes needed to completely enforce security policy between two systems. There is a set of attributes that may be necessary, depending on your security policy, to enforce your policy. The communicating hosts may agree to send all, some, or none of the security attributes from the following set:

- Sensitivity label
- Nationality Caveats
- Integrity label

- TSIX Session ID
- Clearance
- Process Access Control List (ACL)
- Information label
- Capability set
- Login (audit) ID
- Process ID
- Discretionary ID
- Additional client audit info
- Process Attributes

The protocol used to communicate the attributes between systems is SAMP. This is a header and a list of attributes that is prepended to outgoing data as if it were user data. The TCB at each end of the connection has to put this information on, and pull it off and use it before the "real" data is passed to the user program.

Under Trusted IRIX/CMW, each attribute is represented in ASCII form. The ASCII representation is the same as your label name, for example, *TOP SECRET*. User and Group ID numbers are also an ASCII user or group name. Each attribute has its own defined formats in the SAMP standard.

Trusted IRIX/CMW converts the ASCII representations to a 32-bit token that is prepended to the packet for transfer. The SATMP protocol defines these tokens.

When two systems communicate over the network, they first perform a SATMP handshake. During this handshake, the two systems negotiate which predefined Domain of Translation they have in common for each required attribute in the SAMP protocol. Your *rhost.conf* database specifies which attributes are required of the remote host, and what defaults may be substituted if the remote host cannot supply a particular attribute. Your DOT are established as you build your network's *rhost.conf* database.

The remote host database also allows the system to distinguish which hosts it can use TSIX to communicate with, which have only IP Options (such as CIPSO), which are unlabeled, and what level of trust to give each type of host. For example, you can specify that your system communicate with unlabeled hosts, but for purposes of enforcing policy, treat all communication as if it is unclassified, low integrity, and the user is *guest* with no capabilities. This specification happens in the *rhost.conf* file.

If the two hosts attempting to communicate can not find a common DOT for each attribute or supply appropriate defaults, no communication happens. This protects system security.

A well-planned network implements user ID and group ID numbers centrally, so that they are identical across the entire network. If this is the case, these numbers need not be mapped. Similarly, capabilities and labels should also be identical across your network.

Domains of Translation and Heterogeneous Networks

Within an organization of Trusted IRIX/CMW systems and users, there may exist different subgroups of users and systems that use different sets of label hierarchies and categories. For example, one group may have sensitivity levels that are named *Unclassified*, *Confidential*, *Secret*, and *Top Secret* and categories named after military weapons systems, while another group may have sensitivity levels called *Public*, *Company Confidential*, *Management Only*, *Directors Only*, and *CEO Only*, and may have categories named *Payroll*, *Hardware Engineering*, and *OS Software*.

In a heterogeneous environment, you have several options:

1. Hosts can be unlabeledx Security attributes are implied by the default definition in the *rhost.conf* file.
2. Hosts may have IP security options only, such as CIPSO. In this case the Sensitivity Label is described by the Domain of Interpretation. There is also a Trusted IRIX/CMW IP option that sends integrity label and user id number information.
3. Hosts may be TSIX hosts. In this case the host supports sending any of the standard list of attributes.
4. Not all attributes are supported on all systems. Missing attributes are implied by the use of defaults by the receiving host.

There is one DOT for each security attribute, and the DOT describes the interpretation of the attribute. In the case of Sensitivity labels a DOT is a mapping similar in concept to the Domain of Interpretation.

A DOT means other things for other attributes. The account called *darius* on one system can be matched to *dcouch* on another. user ID numbers and GID numbers usually are not mapped, because large installations usually ensure that user ID numbers are identical on all systems.

The other special case is capabilities. These are usually vendor specific mappings in a heterogeneous environment. Because the implementation of privilege is highly variable, mappings are frequently difficult or impossible, and best left to defaults.

DOI/DOT Restrictions Under Trusted IRIX/CMW

Trusted IRIX/CMW implements only one Domain of Interpretation (DOI), DOI 3, but multiple DOTs between TSIX-supporting hosts. The mapping tables for the single DOI are identical to the contents of the files in the directory */etc/mac_label*. The DOI identification number is specified for each interface using the *iflabel* program. Because Trusted IRIX/CMW supports only one DOI, the same DOI identification number should be used on all CIPSO and SGIPSO interfaces.

Configuring a DOI Under Trusted IRIX/CMW

If systems in different groups use the same category and level names but have different number values for these names, label confusion can occur. For example, consider Table 4-1:

Table 4-1 Label Confusion

Label Name	Group A Number	Group B Number
Unclassified	0	10
Confidential	10	20
Secret	20	30
Top Secret	30	40

Obviously, if the computers of these two groups were to begin to interoperate (for example, a gateway was added between their respective networks), then a computer in group A could send “Secret” data with the number 20, and when it arrived at a group B computer it is understood as “Confidential” data. This communication causes a writedown (downgrading) of that data. This type of situation is expressly prohibited for trusted systems.

Continuing the example, consider a group C, which uses the names Public, Company Confidential, and so on according Table 4-2. If this group adds a gateway to group A, there is a label mismatch. Table 4-2 depicts the mismatch:

Table 4-2 Label Mismatch

Number	Group A Name	Group C Name
0	Unclassified	Public
10	Confidential	Company Confidential
20	Secret	Management only
30	Top Secret	Directors only

There is a mismatch problem if these two groups begin to interpret the numbers coming from the other system according to their own tables. Although there is a surface similarity between the numbers used to represent the labels, neither group wants its top-level information to be available to the other group. For more information on label dominance, see the *dominance*(5) reference page.

There may also be an overlapping situation, where some systems are able and cleared to understand multiple sets of information. For example, consider a computer that is allowed to use information according to the following hierarchy. This system is able to receive and send data securely among groups A, B, and C:

```
Public
Unclassified
Company Confidential
Confidential
Management Only
Secret
Directors Only
Top Secret
CEO only
```

To avoid writedown or overlapping, each group must have a set of mapping tables (one for sensitivity level, one for sensitivity category, one for each integrity grade and one for each integrity division). This set of tables facilitates the translation of label names onto numeric values and conversely of number values onto label names. Any such set of tables may be called a map.

Under Trusted IRIX/CMW, the actual map tables are placed into the file */etc/mac*. The DOI identification numbers are configured onto the interfaces using the *iflabel* program. For convenience, the file */etc/config/netif.options* contains the *iflabel* commands that are routinely run at system start-up. The *netif.options* is a good place to put the DOI identification numbers.

The inetd Network Service Daemon

Most of the network services offered by a host system are offered through a program called *inetd*. This daemon is run automatically during system startup if the network has been turned on with the *chkconfig* command. The specific command used is

```
chkconfig network on
```

Rather than having each system service run continuously, *inetd* offers the services of these daemons on their behalf. When a request for one of the offered services arrives and is received by *inetd*, it spawns a child process according to the service requests (for example, *inetd* spawns *rlogind* for an *rlogin* request).

Typically, each new request for a particular service results in a new instance of the daemon that provides that service being spawned. That daemon remains in execution for the duration of the request and then terminates.

The Trusted IRIX/CMW version of *inetd* offers a special service. It can offer any of its services at all labels.The *inetd* daemon can receive requests for any service at any label. Under Trusted IRIX/CMW, *inetd* can, for example, receive *rlogin* requests at any label from *dblow* to *msenhigh/mintlow*, and *inetd* spawns the appropriate daemon at a label matching that of the incoming request. Because of this service, most programs that run as spawned children of *inetd* do not have to be modified in any way to understand labels; they do not have to be made “label cognizant." These programs, running as children of *inetd*, are invoked at the correct label in each instance by *inetd*.

Although they do not have to be made “label cognizant,” most of the child processes of *inetd* run as root, and have root access to the privilege of programs. All programs run by *inetd* must be within the TCB. The fact that a program or file is part of the TCB is signified to *inetd* by the program or file bearing the *dblow* label. The *inetd* daemon cannot (due to MAC) spawn a child process unless the program file for the process is labeled *dblow*. For example, the file */usr/bsd/rlogind* is labeled *dblow*, so it can be run by *inetd*.

If a system administrator takes untrustworthy software, such as programs taken from other source bases or from other operating systems (for example, standard IRIX) and labels it *dblow*, thereby making it a part of the TCB, that system administrator is violating the integrity of the TCB and can open the system to security violations.

Not all network services are offered by *inetd*. Several notable exceptions are the *portmap* service offered by the *portmap* program, NFS, the X Window server, and *sendmail*. The *sendmail* and *portmap* services have been made label cognizant and offer services to clients at all labels. NFS uses the *nfsd* and *biod* servers. These are kernel processes and as such they are also a label-cognizant part of the Trusted IRIX/CMW kernel.

Miscellaneous Trusted Network Information

The following sections contain information useful to the administrator of a trusted network:

- The /etc/hosts.equiv and $HOME/.rhosts Files
- Maintaining the System Audit Trail
- NFS Under Trusted IRIX/CMW
- Using Electronic Mail

The /etc/hosts.equiv and $HOME/.rhosts Files

Under Trusted IRIX/CMW, only the first field of the */etc/hosts.equiv* and *$HOME/.rhosts* files is relevant to the system. The second field is ignored as a comment. This behavior places a restriction on the *rsh* and *rlogin* programs, which do not allow unchallenged access (access without demanding a password) unless the remote user name and user ID are exactly identical to the local user name and user ID. If a different name or user ID is used, the user is prompted for a password that authenticates the user's identity in the usual manner.

Maintaining the System Audit Trail

For a network of several workstations, the audit trail can accumulate a considerable amount of data. The current estimate of audit information is that if all auditing is enabled, a typical workstation can expect to generate 10,000 bytes per second of audit information. Most systems are not able to store this information adequately on the computer's hard disk for a great length of time. It is imperative that a multi-workstation Trusted IRIX/CMW network have a secure and thorough electronic backup plan.

It is also imperative that the individual systems keep their system clocks synchronized. This is accomplished with the *timed* daemon. One system is the designated master time server. All the other systems on the network synchronize their clocks with the server.

The reason that individual stations must keep their clocks synchronized is so that the combined system audit trail is accurate. If attempts to compromise system security are happening, the auditor needs to know the exact sequence of events on each system, with respect to every other system on the network. The way to maintain order in the audit logs is by running *timed*.

Complete information on running *timed* is found in the *timed*(1M) reference page.

NFS Under Trusted IRIX/CMW

Trusted IRIX/CMW provides full support for the NFS version 3 filesystem-sharing software. NFS is an optional software product. For a complete discussion of NFS, see your standard IRIX NFS documentation.

The base NFS protocol does not support MAC labels, ACLs, or capability sets on files. This information is transferred over the trusted network using SAMP and SATMP as with any other network traffic, as described above. The *mount*(1) reference page describes how to mount filesystems using NFS.

Using Electronic Mail

Electronic mail service is provided under Trusted IRIX/CMW. There are some restrictions, however, to the use of mail.

Labeled Mail

As with all other data objects on the system, electronic mail is labeled. When someone sends mail, the mail message inherits the sender's current label. According to the rules governing Mandatory Access Control, the recipient of the mail must be running at a correspondingly equal or dominant label to read the mail sent. If the recipient is running at a lower label, he or she must change labels to an appropriate label to be able to read the mail. If the sender is running at a label dominating the recipient's maximum label, the recipient cannot read the mail or even be notified of its existence until his or her available labels are expanded to include the label of the mail.

MAC protection extends into replying to mail as well as reading it. If someone receives a message from a user at a lower label, the recipient is able to read the mail, but if the recipient responds to the mail, the system attaches the higher label of the recipient to the response. If the original sender is not cleared for the recipient's label, he or she cannot read the response. This policy prevents accidental "writedown" of information from the higher label to a lower label.

To send mail across a gateway onto a unlabeled network, a user must send the mail at the assigned label for the unlabeled network. All mail traffic arriving from an unlabeled network is, of course, labeled with the label assigned to that network.

Using sendmail With Trusted IRIX/CMW

The mail system is driven on each system by a daemon program called *sendmail*. The *sendmail* program is a mail routing clearinghouse. When you use your preferred mailer, such as *Mail*, *sendmail* takes the message you have created and prepares it for the destination system. Then *sendmail* calls the appropriate utilities to deliver the mail locally or queues it for transmission. The *sendmail* daemon has been made label cognizant for Trusted IRIX/CMW. For more general discussion of *sendmail*, refer to *IRIX Admin: Networking and Mail*.

Chapter 5

Administering Access Control

Access control under Trusted IRIX/CMW has been described in general earlier in this guide. This chapter contains a detailed description of the mandatory and discretionary access control mechanisms. The Mandatory Access Control mechanism is the system of labels and clearances that enforce Mandatory Sensitivity and Mandatory Integrity of system objects. The Discretionary Access Control mechanisms are the standard system of file permissions, and the use of Access Control Lists on files and directories.

Sections in this chapter include:

Mandatory Access Control

Mandatory Access Control (MAC) is the most visible feature of CMW security. Under MAC, the system associates a *label* with each process, user, file, or device known to the system. Based on the relationship between the labels of two or more of the items, Trusted IRIX/CMW makes access control decisions.

Within MAC, two separate mechanisms control user access to files and programs. One is Mandatory Sensitivity (MSEN) and the other is Mandatory Integrity (MINT). MSEN is simply the level of protection that a file or object needs, and the corresponding clearance of a user to view or use that file or program. MINT is the indication of how much trust the system has that the file or program is secure or valid, not corrupted or suspect. MINT declares the integrity of the file or program. For example, if an instruction to transfer a large amount of money is sent to the computer, the user wants to know that the message was trusted to be valid, and not a security breach being exploited. MINT assures that users with access to protected data have only trusted tools of high integrity with which to work.

Each label for a system subject (such as a user or process) or system object (such as a file or hardware resource) contains several components. These components are the sensitivity level, integrity grade, and possibly a number of sensitivity categories and integrity divisions. Sensitivity categories divide information into working sets. Information in one category is presumably unrelated to information in any other category. Any subject or object may be in multiple categories or have no associated categories. Similarly, integrity divisions classify different types of information based on decisions to trust the integrity of the information. The sensitivity level of a user determines what level of sensitive information he or she is allowed to use. Conversely, the integrity grade determines how trusted the information must be in order for the user to see it. The higher the integrity requirement, the more trusted the information must be.

For *subjects*, which are active entities such as users and processes, the sensitivity level, sensitivity categories, integrity grade, and integrity divisions together are called a *clearance*. Because subjects usually access and modify objects, subjects require clearance to perform those tasks. *Objects*, on the other hand, have classifications of sensitivity, integrity, divisions, and categories. The clearance of a subject must be at least equal to the classification of an object for MAC to allow the subject access to the object.

When a user logs in, the shell process created by the *login* program inherits the label that the user entered during the login process. The maximum and minimum clearances of a user are stored by the system in the */etc/clearance* file. A user can log in at any clearance up to his or her maximum and down to the minimum. The login shell and any subsequently created processes inherit the login clearance as their label.

MSEN categories and MINT divisions within a label define the nature of the subject or object. For example, a user with the highest sensitivity in research and development does not necessarily have a need to see personnel or accounting information. Therefore, MAC allows you to create categories of information. A high clearance in one category does not allow access to information in other categories.

Each object can be defined in the label as belonging to a number of categories, or to no categories. Each user has a number of categories or no categories in their label. Each new process inherits the label, including all categories of the invoking user. Also, you can define MINT divisions to create a set of tools of known high integrity and limit their use to certain users. A user must have the same or a superset of the MINT divisions of an object in his or her label in order to use the object. All other requirements of sensitivity and general integrity still apply.

The concept of label domination and equivalence is central to MAC. If a subject's clearance is greater than an object's classification and the integrity of the object is good enough for the subject, the subject is said to dominate the object. If the clearance and classification are equal, the labels are said to be equal. A subject must be at least equal to or must dominate an object in order to access it. For more information on label domination and equivalence, see the dominance(5) reference page.

When you add categories to MAC, you affect the usual order of dominance of your security classifications. In order to dominate, a 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 object, 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 object. Table 5-1 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 5-1 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 object 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 subject 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

Types of Labels

The following sections describe the various types of security labels used under the Trusted IRIX/CMW system.

Trusted IRIX/CMW Default Labels

Your Trusted IRIX/CMW operating system comes with a small set of predefined labels. Do not delete these labels for any reason, because all but one of them are administrative. The defined labels are shown in Table 5-2.

Table 5-2 Trusted IRIX/CMW Default Labels

Label	Purpose	Level/Grade
dblow	for TCB changes only	msenlow/minthigh
dbadmin	for editing system data files	msenadmin/minthigh
userlow	for creating new user files	unclassified/highestgrade
equal	for /dev/null	msenequal/mintequal
binary	alias for msenlow/mintbiba	msenlow/highestgrade

Additionally, various default MSEN levels and MINT grades are defined in the */etc/mac* file. You may use these hierarchical components and the default categories and divisions to make more labels or you can define your own.

Equal (Wildcard) Labels

Equal (or wildcard) labels are a special type of label reserved for use by the system and by the system administrator. An *equal* label is sometimes referred to as a *wildcard* label, because it always compares as equal in any label comparison. For example, a user running at *userlow* perceives the *equal* label as also being at *userlow*, while a user running at *userhigh* perceives that label as *userhigh*.

Administrative Labels

The administrative labels are *dblow* and *dbadmin*. These labels are not considered directly comparable with ordinary user labels. They are arbitrary definitions of the lowest and highest possible sensitivities on the system. These labels are generally reserved for system objects and administrative accounts. *dblow* files are considered to be part of the TCB, and they are of low sensitivity and the highest integrity. Thus, they are dominated by all labels and accessible to all users. For example, the */bin/cat* program is available to all labels. *dbadmin* files are of the highest integrity and also of the highest sensitivity found on the system. These files contain data that must not be divulged or compromised. For example, the */etc/shadow* file, which contains each user's encrypted password, is not available for general perusal.

No user should be able to log in at either of these labels, although exceptions are made for user accounts belonging to system administrators.

User (TCSEC) Labels

Ordinary user labels (also called TCSEC label types, after the NSA Trusted Computer System Evaluation Criteria) are those that the user sees in day-to-day work. Trusted IRIX/CMW is shipped with only one user-level label configured, *userlow.* As your user label library grows, remember to keep the hierarchy of labels clear, consistent, and easy to understand.

User labels always dominate, but are never equal to *dblow*. No user label can dominate *dbadmin*.

Batch processing jobs (such as those submitted through *at*, *batch*, and *cron*) may be submitted at any label for which the user is cleared, and they are run with that label. Entries in *crontab* files are also allowed at any label for which the user is cleared, with a separate file being maintained at each label requested.

Changing User Default Labels

If the user's default label is changed while there are scheduled tasks pending, the tasks will be run at the new default label, not at the label at which they where submitted.

Before changing a default label, the system administrator must verify that either the user has no outstanding requests or that the requests are appropriate at the user's new default label. The system administrator can find the background task request information in the */usr/spool/cron/atjobs* and */usr/spool/cron/crontab* files, as described in the cron(1M) reference page.

Complete information on *at*, *batch*, and *cron* is available in their respective reference pages, and in the guide titled *IRIX Admin: System Configuration and Operation*.

Multilevel Labels

It is possible to hold a multilevel label. You can use the following commands to change your current label to multilevel:

```
newlabel -m
su ,username. -m
```

The advantage to multilevel labels is that directories can also be made multilevel. A multilevel directory (or, sometimes, *moldy* directory) places files of different labels into multiple hidden subdirectories. Each subdirectory bears the label of the files in that subdirectory. Thus, process A with label *userhigh* sees a different listing of the contents of the *mld* from process B with label *userlow*. However, neither process sees the subdirectory structure. Each process sees only files with the same label as the process in the *mld*. Once your label is multilevel, you can see the multilevel directory structure, but the rules of dominance are still in effect. You cannot see the contents of a subdirectory whose label you do not dominate, though you can see that the subdirectory exists. To create a multilevel directory, use the *mkdir* command and then use the *chlabel* command with the **-m** flag and the name of the directory; the directory becomes multilevel.

Three types of *multilevel* labels are available: *msenmldhigh*, *msenmld*, and *msenmldlow*. *msenmldhigh* is a multilevel *msenhigh* clearance to give MSEN (though not necessarily MINT) dominance over all files on the system. *msenmld* labels at other MSEN levels are subject to the rules of dominance. *msenmldlow* is a multilevel *dblow* label for working with the TCB. You can log in with a multilevel label of one of these types, or you can log in at *msenhigh/mintequal*, *dblow*, or any label and use *newlabel -m* to make your label multilevel.

Working With Labels

The following sections detail how to examine and manipulate labels on your Trusted IRIX/CMW system:

- Checking Labels
- Changing Object Labels
- Changing Process sLabels
- Creatign New Label Names
- Deleting a Label

Checking Labels

Frequently, you will find it necessary to check the label of a file or directory, or perhaps the labels of all the files in a *moldy*directory. For this purpose, Trusted IRIX/CMW supports the **-M** flag to *ls*. The *ls -M* command lists files according to the usual behavior of *ls*, except that the human-readable names of the labels attached to the files or directories are displayed as well.

The *id* command displays the calling process ID number and name. It also displays the group ID and name. If the real and effective IDs do not match, both are printed. When invoked with the **-M** option, *id* reports the MAC label at which the invoking process is running.

Using attrinit to Clean Up Label Corruption

If you believe you have experienced corruption of some labels , you can use the *attrinit* command to restore your system labels. See the attrinit(1) reference page for more information on *attrinit* command.

The */etc/irix.mac* file is used with the *attrinit* command as follows:

1. Log in as root and change directories to the / directory.
2. Enter this command:

```
attrinit -script=etc/irix.mac
```

It takes several minutes while your labels are restored.

Changing Object Labels

You can change the label of an object with the *chlabel* command. Be aware that you must have access to the object before giving the command. You cannot use *chlabel* to change the label of an inaccessible object. The new label cannot be less sensitive or of higher integrity than the old label. Additionally, the current label of the object must be equal to that of the process attempting the change. The **-m** flag to *chlabel* changes the label of a directory to multilevel.

The system administrator may set a file or directory to any label; you must have sufficient privilege. If auser accidentally changes the label on a file and can no longer access the file, the system administrator must downgrade the file for the user. The system administrator is also the only user who may set an *equal* label. For complete information about the *chlabel* command, consult the chlabel(1) reference page.

Changing Process Labels

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. The processes you may run include opening a new shell window and using the command *su -M*.

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.

You may also use the *su* command with the **-M** option to execute a command at a higher label, provided you are cleared for that label, or have the password of an account that is cleared for that label. See the *su*(1M) reference page for complete information.

Creating New Label Names

From time to time, it may become necessary to create new label names on your system, to adapt to changing usage and new projects. You also must create your starting set of labels when you install your system. The */etc/mac* file defines the sensitivity clearances and categories and the integrity grades and divisions. This file contains a "starter set" of predefined labels and label components for you to use. As you add your own labels to this file, remember that in no instance should you delete any of the distributed label components. Trusted IRIX/CMW depends on many of these defined labels and components. Also, remember as you edit that everything you put in the file is case-sensitive. That is to say, the system differentiates between uppercase and lowercase letters in the names of the items. Also, each new label name must be unique. For example, you cannot use the name "good" for both a sensitivity level and an integrity grade.

To add new label names, edit the */etc/mac* file at *dblow.* The format is

```
name:type:value
```

The following table is a short summary of the types and values in the */etc/mac* file:

Table 5-3 Types and Values in the /etc/mac File

Type	Value	Definition
gradenames	*numeric value*	The *gradenames* type defines integrity grades. The higher the numeric value, the greater the integrity of the grade. You must decide where you wish to position the new grade in your integrity hierarchy by examining the numeric values of the existing integrity grades and assigning the new number at the appropriate level.
divisionnames	*numeric value*	The *divisionnames* type defines integrity divisions. For divisions, the numeric value of the new definition is arbitrary and for identification only. You need only make certain that the new number is not already in use. The values 0-99 are reserved for system use.

Table 5-3 Types and Values in the /etc/mac File

Type	Value	Definition
levelnames	*numeric value*	The *levelnames* type defines sensitivity levels. The higher the numeric value, the greater the sensitivity of the level. You must decide where you wish to position the new level in your sensitivity hierarchy by examining the numeric values of the existing sensitivity levels and assigning the new number at the appropriate level. Note that if you are using CIPSO Type 1 as your networking environment, only categories 0-63 can be transmitted.
categorynames	*numeric value*	The *categorynames* type defines sensitivity categories. For categories, the numeric value of the new definition is arbitrary and for identification only. You need only make certain that the new number is not already in use.

Deleting a Label

Never delete a label name once it has been used on your system. Change all files that have the label to another label and remove the label from the ranges of all your users. Then, comment out the entry in the */etc/label/labelnames* file by placing a pound sign (#) at the beginning of the definition line. In this way, no one can use the label, just as if it had been removed, but a record remains that the label existed. This is necessary to prevent accidental reuse of the label for another purpose. If the name of a sensitivity level, category, integrity grade, or division is reused, a declassification of information could result. For example, if a sensitive file is left unchanged after a label is removed, and the label name is reused, that sensitive file is available to users not cleared for the information.

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 added feature of Access Control Lists (ACLs) is implemented in IRIX. 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. Further, the user may set an ACL for any file or directory. ACLs are discussed in the section titled "Access Control Lists."

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. Thus, 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.

Trusted IRIX/CMW File Permissions

Trusted IRIX/CMW divides permissions into three categories, and users into three relationships. The three categories of permissions are *read*, *write*, and *execute*. They are denoted as "r" for read, "w" for write, and "x" for execute.The three relationships are the owner of the file, the owner's user group, and every other user. If you get a long listing of a directory, you see that the permissions field for each file in the directory looks something like this:

```
-rwxrwxrwx
```

Note that the line of permissions has the string `rwx` repeated three times. The first instance of `rwx` applies to the file owner, the next instance applies to the group members, and the third applies to all other users on the system. The example above shows full permissions. A more restricted permission set might look like this:

```
-rw-r--r--
```

To get a long listing of file permissions, enter this command at your system prompt in any directory:

```
ls -l
```

Along with the permission information, the *ls -l* command lists the owners of the files, the size of the files, and the date they were last modified.

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.

Each character is separately significant in the permissions listing. Starting at the left, the first character is a dash. A dash in any other position 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 describe whether the file is a file or a directory. If it is a directory, a "d" appears in that space. Other characters in this place indicate that the file is a pipe, a block or character special device file, or other type of file. See the *ls*(1) reference page.

Directory Permissions

Directories use the same permissions as files, but 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 set of three places after the leftmost place in the permissions field describe the permissions for the owner of the file. Here is an example ofa long listing for a file:

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

The file is not a regular file, so the leftmost 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 set of three spaces describe 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
```

Any member of the group `grp` could read or execute the file, but 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 files that are owned by one person, but any group member can work on them.

The third set of spaces provides for all other users on the system and is called the public permissions. 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 a 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 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 in this directory have varying permissions. Some are restricted to 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 is executable by any user.

Changing Permissions

You change the permissions on a file by using 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 as follows:

`chmod 600` *filename*

`chmod 700` *dirname*

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 as follows:

`chmod +w` *filename*

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

Setting Permissions With umask

You can decide what default permissions your files have by placing the *umask* command in your *.cshrc*, *.profile*, or *.login* file. There is a default *umask* setting for the entire system in the */etc/profile* and */etc/cshrc* files. By changing the setting of your umask, you can alter the default permissions on your files and directories to any available DAC permission. See the *umask*(1) reference page for more information.

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, you might want the personal files you keep in your home directory to be private, and if you set your umask to allow group read and write access, any member of the group can access your personal files. But 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 no other user would be allowed into the directory.

You can also use the Trusted IRIX/CMW utilities 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 to any files or directories you indicate every time you log out. For example, say you have three directories, called *personal*, *letters*, and *budget*. You can set up a *.logout* file in your home directory with commands to be executed each time you log out from the system. The following commands, placed in the *.logout* file, will prevent access to the three example directories for anyone but you:

```
chmod 700 budget personal letters
chmod 600 budget/* personal/* letters/*
```

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

```
umask 006
```

This makes every file you create have the following permissions:

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

With your umask set to 006, directories that you create have the following permissions:

```
drwxrwx---
```

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

Access Control Lists

An Access Control List (ACL) works in the same way as standard file permissions, but it allows you to have a finer level of control over who may access the 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 to subdirectories 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 allowed by the access ACL and by MAC to gain access successfully.

Hereafter in this section, directories are treated as files, and where the term file is used, consider that it also applies 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 *ls -D* command, as shown in this example:

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

This produces output similar to this:

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

This 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/write permission for the user account ernie. The specific format of an ACL entry is discussed in the section titled “Long ACL Text Form.”

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

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

An ACL consists of a set of ACL entries separated by commas. 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. To read an ACL from an object, a process must have read access to the file. To create or change an ACL, the process must own the file.

ACLs have long and short text forms. 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 ACL Text Form

The long text form is used for either input or output of ACLs and is set up 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 improves readability.

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	Access granted to either the file owner or to a specified user account.
group	Access granted to either the file-owning user group or to a specified user group.
other	Access granted to any process that does not match any user, group, or implementation-defined ACL entries.
mask	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	User account name or a user ID number.
gid	User group name or a group ID number.
empty	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 discretionary access permissions field may contain each of the following characters:

r	Read access
w	Write access
x	Execute access
-	Placeholder

Any or all of these may be replaced by the no-access dash(–).

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 numeric *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 *umask* 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 *umask* entry, then the output text form for that entry must be displayed as described above followed by a crosshatch (#), 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,group::rwx,other::rx,mask::rx,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/write 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 ACL Text Form

The short text form is used by the *chacl*(1) command for input of ACLs, and is set up as follows:

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

The abbreviations are as follows:

u	User
g	Group
o	Other
m	Mask

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

Using ls -D and chacl

You can use the output from the *ls -D* command as the input to *chacl*. This is convenient for situations where you wish to duplicate a complex custom ACL onto a new file in a directory that does not use the complex ACL as the default.

Consider the command:

```
ls -dD testdir
```

It produces the following output:

```
testdir [u::rwx,g::r-x,o::--x/u::rwx,g::r-x,o::---]
```

Capability Assignment

The capability-based privilege mechanism, described in Chapter 2, assigns capabilities to the system administrator (*root*) and to the auditor (*auditor*) as requested at login based on the contents of the user capability database file (*/etc/capability*). These capabilities determine the amount of privilege granted by the capability-based privilege mechanism to the system administrator and auditor. (For more information on the format of the user capability database, see the capability(4) reference page.)

Note: In the augmented superuser privilege envirionment, the system administrator has unlimited privilege regardless of the contents of the user capability database, though the auditor is still constrained.

It is inappropriate to grant capabilities to users other than root and auditor. Software used by ordinary users is automatically granted the capabilities it needs based on file capabilities; a user with capabilities beyond this is some form of administrator. The IRIX administrative model does not support administration by users other than root and auditor, so assignment of user capabilities to other users is not supported by IRIX or Trusted IRIX/CMW systems.

In addition to user capabilities, the IRIX system employs capabilities assigned to files. For details on how these work, see the capabilities(4) reference page. File capabilities can be viewed by using the *ls -P* command and changed by using the *chcap* command.

The capability sets assigned to files in the IRIX system and related products are designed to establish appropriate privilege for correct system operation. Changing the file capabilities on an IRIX command or Silicon Graphics product may degrade system functionality or compromise security. You must take care when assigning or change file capabilities to avoid this risk.

If you believe that the file capabilities on your system have become corrupt, you can use the *attrinit* command to restore file capabilities to their original settings (see the attrinit(1) reference page). The */etc/irix.cap* file is usedwith the *attrinit* command.

Follow these steps:

1. Log in as *root*.
2. Change your directory to the / directory.
3. Enter the following command:

```
attrinit -script=/etc/irixcap
         -verify=installed
```

Your capability integrity will be restored. The process may take a few minutes.

Chapter 6

Administering the System Audit Trail

The *system audit trail* is a feature that allows the system administrator or auditor to review a record of all system activity. The ongoing record shows general trends in system usage and also violations of your system use policy. For example, any unsuccessful attempts to use system resources can be recorded in the audit trail. If a user consistently attempts to access files owned by other users, or attempts to guess passwords, this can be recorded in the audit trail. The system administrator or auditor can monitor all system activity through the audit trail.

The system audit trail is described in the guide titled *IRIX Admin: Backup, Security, and Accounting*. A chapter in that book provides information on the differences in auditing between standard IRIX and Trusted IRIX. All references to Trusted IRIX/B in that chapter are valid for Trusted IRIX/CMW.

The *satconfig* utility is documented in the *IRIX Admin: Backup, Security, and Accounting* volume, but is not present in Trusted IRIX/CMW. Use the *sat_select* utility instead for better security.

Topics described in this chapter include:

Audit Events in Trusted IRIX/CMW

The following audit events are specific to the Trusted IRIX/CMW system:

sat_access_denied
: EAccess to the file or an element of the path was denied due to enforcement of MAC or DAC permissions.

sat_ae_audit
: Events from the SAT daemon.

sat_ae_lp
: Events from the print daemon.

sat_bsdipc_mac_change
: The user changed the MAC label on a socket.

sat_bsdipc_dac_change
: The ACL or user ID of a socket was changed.

sat_bsdipc_dac_denied
: A packet could not be delivered to a socket because of DAC.

sat_bsdipc_if_invalid
: Attempt to change MAC labels was disallowed for lack of MAC permission.

sat_bsdipc_if_setlabel
: The MAC labels on an interface structure were changed.

sat_bsd_if_setuid
: A change was made to an outgoing socket user ID.

sat_bsdipc_rx_missing
: A packet was received on an interface with a missing or damaged MAC label.

sat_bsdipc_rx_range
: A packet was not received due to MAC violation of label range.

sat_bsdipc_tx_range
: A packet was not sent due to a MAC violation.

sat_bsdipc_tx_toobig
: A packet was not sent because the MAC label was too large for the IP header to contain.

sat_proc_acct
: Extended process accounting information events.

sat_proc_own_ext_attr_write
: The calling process's MAC or capability state was changed.

sat_session_acct
: Extended session accounting information

sat_svr4net_create
: An svr4 network connection was made.

sat_svr4net_address
: An svr4 network attribute was set.

sat_svr4net_shutdown
: An svr4 network connection was ended.

sat_sys_note
: System internal status data.

Auditing Unexpected Use of Privilege

Trusted IRIX/CMW has policies implemented that are not present in standard IRIX. Because of this, the unexpected use of privilege is of special concern to the auditor of a Trusted IRIX/CMW system. Every interpreted audit record contains a line beginning with the keyword "Outcome."The field following this keyword can be equal to one of "Success," "Failure," or "Success due to privilege." The last case indicates that the user made a system call that would have failed except that superuser privilege was invoked to assure its successful completion. This is not necessarily a security violation or an unexpected use of system privilege. It is perfectly normal to see these outcomes.

When an ordinary user runs a program that contains code that uses system privilege, "success due to privilege" outcomes are generated. A good example of this kind of program is *passwd*. An ordinary user generates a record of this type simply by changing the password on his or her account. Records of this type are also generated when users use capabilities to edit system files.

One type of record to look for is an instance where the "SAT ID" or "Effective ID" field is different from the "User ID" field. This occurs when a user executes */bin/su* to gain privileges or otherwise promotes the privilege level of a session. In most cases, this is not a security violation, since the capability is necessary to successfully complete the */bin/su* command.

An instance of using superuser privilege, though, is always worth examination in the audit trail. When you encounter an instance where a user has promoted his or her login session, you should check to see that the user is authorized to have the capability. If not, check whether the user indeed executed the */bin/su* command, or if he or she promoted the privilege of the session by some other means, such as a Trojan Horse *setuid* shell command.

Whenever a user promotes the privilege of his or her login session, the auditor should also make a routine check of what actions the user took while the privilege was promoted.

Chapter 7

Administering Identification and Authentication

In order to ensure that the users on your system are the same people who have been entrusted to use it properly, Identification and Authentication (I&A) requirements have been implemented. Further, in the unlikely event that an individual user breaks a security policy, that user must be held strictly accountable for his or her actions. Holding the owner of a user account responsible for the actions taken with that account is reasonable only if steps have been taken to ensure that the individual using that account is in fact the individual assigned to the account. B1-level systems are required to implement certain facilities to ensure that users are adequately identified and that they authenticate themselves to the system with a password. To log in, the user must know:

- a valid login name for the system
- the password associated with that login name

Because these items are all that is needed to gain access to a trusted system, these pieces of information are closely guarded, and created and maintained according to strict procedures outlined in this chapter. Of the two items of information, the most crucial is the account password. The login names are known to many people, or can easily be determined. It is possible to log in without specifying a label if the default label has been set, but the password is absolutely necessary. If a password is compromised or stolen, all information that is available to the user associated with the password is also compromised.

Sections in this chapter include:

Administering Passwords

Passwords are the only mechanism available to authenticate your users. Once Trusted IRIX/CMW has accepted a user's password as valid, that user has access to all files available at his or her clearance for the duration of the login session.

The dangers involved when passwords are compromised cannot be overstated. An intruder can access all files available to the user at any time. Other features of Trusted IRIX/CMW make it likely that an intruder would be swiftly identified and locked out, but a tremendous amount of damage can take place in a short time if the accountability and Identification and Authentication procedures are not followed.

Many features taken for granted in standard IRIX are restricted in Trusted IRIX/CMW. In the area of user passwords, there are facilities to force the user to choose a system-generated password (which is random and difficult to guess). The length of time that this password is valid can be specified (with both a minimum and maximum lifetime), and the encrypted password is not visible to users. When an encrypted password is visible, a potential intruder may copy it and attempt to break the encryption.

Also in Trusted IRIX/CMW, if you choose to allow users to select their own passwords, a strict set of checks are performed on the passwords to disallow passwords without enough variation in the characters used. For example, all passwords must use a combination of letters, numerals, and control characters.

Password Aging

Trusted IRIX/CMW supports password aging. Password aging is defined as being able to set a minimum and maximum lifetime for passwords. Standard IRIX also supports this feature, and it is described in detail in the guide titled *IRIX Admin: System Configuration and Operation*.

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 interloper has to crack the password. By enforcing a minimum password lifetime, you prevent lazy users from simply changing their password briefly and then returning to their usual password immediately.

If a user does not change their password within the specified time period, the user is forced to change the account password when they next try to log in. Any user can place the following line in *.login* or *.profile* to notify them when password expiration is imminent:

`showpwage` *username*

By default, *showpwage* notifies the user only 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 or system administrator. When the account is locked, the encrypted password is replaced by this string:

```
*LK*
```

To unlock the account, the system administrator uses the *dbedit* utility to remove the string from the password field for that account. Then, the system administrator should force the user to choose a new password by executing this command:

```
passwd -f username
```

Password aging is enforced for a particular user if his or her encrypted password in the */etc/shadow* file is followed by a comma and a non-null string of characters from the 64-character alphabet:

```
. / 0-9 A-Z a-z
```

The first character of the entry, *Maximum*, denotes the maximum number of weeks for which a password is valid. A user who attempts to log in after his or her password has expired is forced to change the password. The next character, *minimum*, denotes the minimum period in weeks that must pass before the password may be changed. If the second character is omitted, zero weeks is the default minimum. *M* and *m* have numerical values in the range 0-63 which correspond to the 64-character alphabet shown above (for example, / = 1 week; z = 63 weeks). If *minimum* = *Maximum* = 0 (derived from the string . or ..) the user is forced to change the password at the time of the next login (and the "age" disappears from the entry in the */etc/shadow* file). If *minimum* > *Maximum* (signified, for example, by the string ./) only the system administrator can change the password. This is often done for accounts that are used for uucp logins. For example, the following command disallows the user from changing the password:

```
passwd -x1 -n10 nuucp
```

For complete information on how to put an age limit on a user's password, consult the passwd(1) reference page.

Administering Password Generation

There are several options available to the security officer when deciding on password generation policy. Trusted IRIX/CMW comes equipped with a default password generation utility, */bin/passwd_gen* (installed as *passwd_gen.off*), but also allows the individual site to install a custom password generating program. If you have a custom password generator you wish to use, you may replace */bin/passwd_gen* with your own program, subject to the following constraints:

- The system administrator is willing to accept the risk of using an unevaluated configuration.
- The new program must be placed in the */bin* directory and renamed *passwd_gen*.
- The owner of the file must be *root*.
- The group of the file must be *sys*.
- The DAC permission of the file must be 555 (-r-xr-xr-x).
- The security label of the file must be *dblow*.
- Your system is no longer running the evaluated software configuration.

Additionally, any custom password generation program must generate a set of passwords on one line, separated by blank spaces.

To allow password selection, simply log in as the superuser and rename the */bin/passwd_gen* file.

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 to obtain a new set of passwords.

If you do not wish to implement password generation at your site, you may remove or rename the *passwd_gen* program. Once this is done, users are free to select their own passwords, subject to the following triviality checks:

- 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 numeric 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 are not allowed. For the purposes of this test, capital letters are assumed to be equivalent to their lowercase counterparts.
- The password must have at least three characters different from the previous password. For the purposes of this test, capital letters are assumed to be equivalent to their lowercase counterparts.

Note that if password generation is in effect, the generated passwords are not subject to the above listed triviality checks because more stringent checks are applied internally.

Password Generator Algorithm

The Department of Energy requirements for password generators state that all trusted site must have an "ADP Security Plan" that describes the method of password selection, the length of password, and the size of the password space. To satisfy this requirement for Trusted IRIX/CMW sites, the following information is included in this section.

The Trusted IRIX/CMW password generator produces passwords of 7 lowercase alphabet characters in length. The Trusted IRIX/CMW password generator attempts to produce pronounceable passwords, so it produces far fewer than the maximum possible number of passwords. The total password space (the total number of passwords that the generator can possibly produce) is 35,431,196 different passwords. This number is computed based on the size and number of phonemes available for combination into words and the method by which they're combined.

Trusted IRIX/CMW limits the total password guessing rate (for all accounts, on all tty and pty ports) to no more than 1 password per second. This guess rate is not user-configurable. Thus, a person who knew what parameters were used by the password generation program, who had unrestricted access to a Trusted IRIX/CMW system, and who was capable of attempting to log in once per second would be able to guess any password generated by that algorithm in 35,431,196 seconds, which is 410 days of uninterrupted guessing. In 41 days of uninterrupted guessing, the person would have a 10% chance of guessing any password. In 35 seconds, the person would have a "one in a million" chance of guessing a correct password for a given account.

Of course, it is extremely unlikely that someone attempting to break into a Trusted IRIX/CMW system would know the parameters used to generate passwords, or have unrestricted access to a well-maintained trusted system, so the rate of guessing would necessarily be much lower. If password aging is implemented, requiring users to change their passwords monthly, the chance of a potential intruder correctly guessing a password is negligible.

The password generator relies on a “pseudo-random” number generator, which is described in the random(3) reference page.

Login Process

The process of logging in to a Trusted IRIX/CMW system is more complicated than meets the eye. Many activities go on within the operating system that are configurable through the *login.options* file.

When no one is logged in to a Trusted IRIX/CMW system, the system displays the CMW login dialog and waits for a user to enter a login name.

1. The trusted path window is displayed on the screen as shown in Figure 7-1, and the trusted path should be initialized "on:"

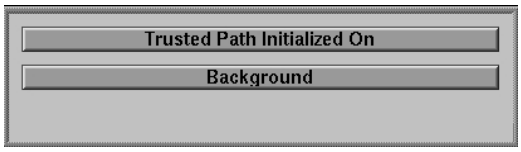

Figure 7-1 Trusted Path Window

2. 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 "You Are On The Trusted Path." Again, if it does not state that you are on the trusted path, call your system administrator.

3. On the CMW Login Dialog window, you should see the prompt as shown in Figure 7-2 (`User Name:`):

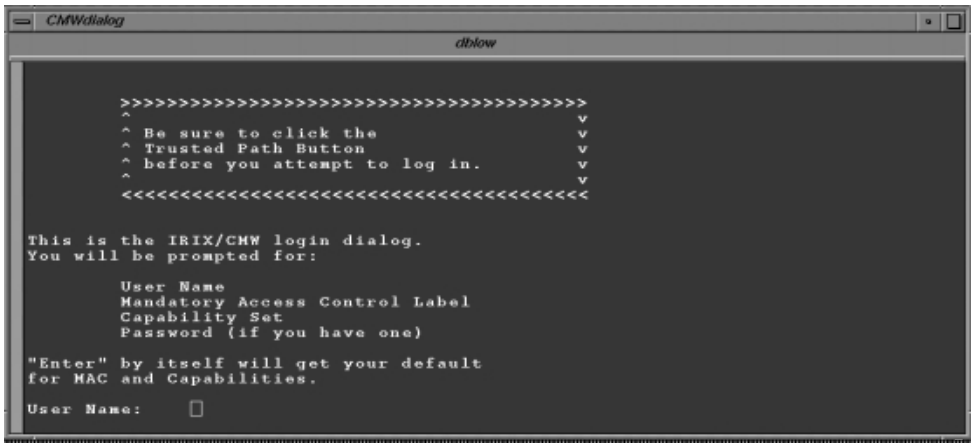

Figure 7-2 CMW Login Dialog Window

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

4. You are prompted for a MAC label:

   `MAC Label:`

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

5. You are prompted for a capability set:

   `Capabilities:`

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

6. You are prompted for your password:

   `Password:`

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

7. If all responses were valid, the user is logged in. If the *login.options* file contains this line:

   ```
   lastlog = 1
   ```

   then the user is notified of his or her last login date and time. This is done so the user can be instantly aware if someone else has logged in to the account since the last login. If the user has never logged in before, the system does not display the lastlog message.

8. The screen clears and the default windows and icons are displayed. The login process is now complete. Your *.sgisession* file executes.

The password step can be eliminated from the login process if the user has no password set and the following string appears in the *login.options* file:

```
passwdreq=0
```

This means that a user who does not have an initial password set does not have to create one or enter any password to log in. Obviously, this is a highly insecure practice, and you should not allow it on your system.

It is recommended that you have *passwdreq* set to 2 on your system at all times. The effect of setting *passwdreq* to 2 is described below. However, even if passwords are not required on a particular system, any user who maintains a password on his or her account must enter it at login time. Regardless of whether passwords are required, the system does not allow access to a passworded account without receiving the correct password.

If the string `passwdreq=1` appears in the *login.options* file, passwords are always required on the system and a user without a password is prompted to choose one immediately. This is the default behavior for the Trusted IRIX/CMW system.

If the *passwdreq* line reads `passwdreq = 2` then a user without a password already set is not allowed access and the system administrator must create a password for the user before he or she can log in.

Assuming that the user enters the correct password, no other user input is required to complete the login process. If the password or any previous entry was incorrectly entered, the system responds with the following message and the login process is aborted:

```
Login incorrect. Try again.
```

If the account is new and has no password and passwords are required, the user sees this prompt:

```
Enter New Password:
```

At this time, the user is forced to enter a password before being allowed to complete the login process. The user is always prompted to re-enter the new password as an error check.

Login Failures

During the login process, login failures are counted and compared against the values set in the *login.options* file. The following line indicates the number of unsuccessful attempts allowed per login port:

```
maxtries = number
```

The default value for this keyword is 5. If the user unsuccessfully attempts to log in 5 consecutive times on the same terminal, the system disallows logins on that terminal for the number of seconds specified in the *login.options f*ile by this entry:

```
disabletime = number
```

The default value for *disabletime* is 20 seconds. The system administrator is exempt from this restriction, because it may be necessary to log in quickly in an emergency.

If the keyword *syslog* is in the *login.options* file with either of the following settings, unsuccessful login attempts are placed in the system log with the date and time:

```
syslog = all syslog = fail
```

login.options File

The *login.options* file allows you to set the following options for all users on the system, as shown in Table 7-1:

Table 7-1 Login Options

Option	Default Value	Meaning
maxtries	5	Maximum consecutive number of unsuccessful login attempts to any login name through the same port. A 0 in this space indicates “no limit.”
disabletime	20	The amount of time in seconds *login* waits before ending the session after *maxtries* unsuccessful attempts.
passwdreq	0	This field indicates whether passwords are required. If this field contains a 0, passwords are not required. If the field contains a 1, you may select a password when you log in if you do not have one. If the field contains a 2, you may not log in without a previously set password.
lastlog	1	This field indicates whether the user is to be notified about the last successful login attempt. A 1 in this field indicates that the user should be notified. If a 0 is present in this field, notification is suppressed.
syslog	all	This field directs the system to log all successful and failed login attempts to the system log. If the value in the field is *fail*, then only failed attempts are logged.

After your installation is complete, you may edit the *login.options* file to enforce your particular system security policies. Before you edit the file, be sure to make a backup copy of the original. If the file is removed, the default values in effect at installation time are used.

/etc/shadow, /etc/clearance, /etc/capability, and /etc/mac Files

When the user logs in, the system encrypts the password and tests it against the encrypted password for the account listed in the */etc/shadow* file. This file is labeled *dbadmin*, not visible to users, thus shielding the encrypted passwords. The */etc/passwd* file is still in existence, though, and still contains all pertinent user information except the encrypted password and label ranges. The *passwd* file also contains information about the minimum and maximum age of that user's password. The */etc/clearance* file contains the security labels allowed for that user.

Chapter 8

Trusted IRIX/CMW System Data Files

The Trusted IRIX/CMW system relies on a number of administrative data files to provide crucial information for the system. It is the job of the system administrator to keep these files correct and up to date. This chapter contains a list of the system data files found under the Trusted IRIX/CMW system and their formats and functions.

The outline format used in this chapter for describing each administrative data file is as follows:

Pathname: The complete pathname of the file.

Description: A complete description of the purpose of the file.

Syntax: The syntax of a record or entry in the file.

MAC Label: The default Mandatory Access Control (MAC) label associated with the file. A MAC label has two symmetric parts; the Mandatory Sensitivity (MSEN) portion and the Mandatory Integrity (MINT) portion, separated by a slash character (/).

DAC Permission: The default Discretionary Access Control (DAC) permissions associated with the file.

Sections in this chapter include:

Home Directory Files

The files described in this section are present in the home directory of each user.

.rhosts

Pathname: *~/.rhosts*

Description: This file contains a list of hosts from which this user is allowed to initiate a remote session without additional authentication.

Syntax: *command hostname username*

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

.sgisession

Pathname: *~/.sgisession*

Description:

Syntax:

MAC Label:

DAC Permission:
-rw-r--r-- (644) root,sys

Files in the /var Directory Structure

/var/adm/OLDsulog

Pathname: */var/adm/OLDsulog*

Description: This file is used for backups of the *sulog* file.

Syntax: Each entry in *OLDsulog* has the following form:

`SU 09/09 10:21 + ttyq2` *invoking user-new identity*

MAC Label: dbadmin

DAC Permission:

-rw------- (600) root,sys

/var/adm/sulog

Pathname: */var/adm/sulog*

Description: This file contains a log of all uses of the *su* command.

Syntax: Each entry in *sulog* has the following form:

`SU 09/09 10:21 + ttyq2` *invoking user-new identity*

MAC Label: dbadmin

DAC Permission:

-rw------- (600) root,sys

Files in the /dev Directory Structure

The following files reside in the special */dev* directory structure. These device files control the physical hardware.

/dev/console

Pathname: */dev/console*

Description: The console provides the operator interface to the system. The operating system and system utility programs display error messages on the system console.

The console is a logical terminal represented by a text window on the graphics monitor.

The evaluated configuration does not support the option of using a serial terminal.

The device special file */dev/console* represents the system console. */dev/console* is the slave side of pseudo-tty (see the pty(7) reference page).

Syntax: Special Device File

MAC Label: dblow

DAC Permission:
crw--w--w- (622) root,sys

/dev/klog

Pathname: */dev/klog*

Description: The */dev/klog* file is the kernel error logging interface. When this device is open, messages printed by the kernel, which normally appear only in the system console window, also are buffered by the *klog* driver. The messages obtained by reading from this driver are the text of the kernel error messages.

Normally, this device is opened and read by *syslogd*, the system logging daemon.

Syntax: Special device file.

MAC Label: dblow

DAC Permission:
crw-r--r-- (644) root,sys

/dev/kmem

Pathname: */dev/kmem*

Description: */dev/kmem* is a special file that is an image of the kernel virtual memory of the computer. It may be used, for example, to examine and even to patch the system memory.

MAC Label: dblow

DAC Permission:
crw-r----- (640) root,sys

/dev/log

Pathname: */dev/log*

Description: This file is a named pipe that is read by *syslogd* as a source of system log messages. If a program writes error messages to */dev/log*, *syslogd* receives the messages and places them in the system log.

Syntax Named pipe.

MAC Label: dblow

DAC Permission:
prw-rw-rw- (666) root,sys

/dev/ptc

Pathname: */dev/ptc*

Description: This file is the master pseudo-terminal.

MAC Label: dblow

DAC Permission:
crw-rw-rw- (666) root,sys

/dev/tty

Pathname: */dev/tty*

Description: This file is, in each process, a synonym for the control terminal associated with the process group of that process, if any.

MAC Label: dblow

DAC Permission:
crw-rw-rw- (666) root,sys

Files in the /etc Directory Structure

/etc/TIMEZONE

Pathname: */etc/TIMEZONE*

Description: This file contains the time zone (for example, EST), the hours of difference between the time zone and Greenwich time zone (for example, 5), and the alternative time zone (for example, EDT). All the information is in one line without any field separators.

Syntax: TZ=*timezone hours_from_GMT daylight_timezone*

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/capability

Pathname: */etc/capability*

Description: This file specifies the system-file editing permissions for each account on your system. This file contains the following information for each account:

name	User's login name—contains no uppercase characters and must not be longer than eight characters.
capabilities	The various capabilities that the user is allowed.

Syntax: The following is a sample *capability* file:

```
root:all+eip:all+eip
sysadm:all=:all=
auditor:CAP_AUDIT_WRITE,CAP_AUDIT_CONTROL,CAP_KILL+eip
dbadmin:all=:all=
ernie:all=:CAP_FOWNER,CAP_SETFCAP+eip
casey:all=:all+eip
```

MAC Label: dbadmin

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/clearance

Pathname: */etc/clearance*

Description: This is the user label file. This file contains the following information for each user:

name	User's login name—contains no uppercase characters and must not be longer than eight characters.
default security label	The default label assigned to the user if no label is specified.
minimum security label	The lowest security label that the user is allowed.
maximum security label	The highest security label that the user is allowed.

Syntax: The following is a sample *clearance* file:

```
root:dblow:dblow...dbadmin
bill:userlow:userlow...userhigh
```

MAC Label: dbadmin

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/cshrc

Pathname: */etc/cshrc*

Description: This file is the prototype .cshrc.

Syntax: This file contains a sample of C-shell initialization commands. It is used as the default set of commands.

MAC Label: dblow

DAC Permission:
-rwxr-xr-x (755) root,sys

/etc/gettydefs

Pathname: */etc/gettydefs*

Description: This file contains information used by *getty* to set up the speed and terminal settings for a serial line. This file supplies information on what the *login*(1) prompt should look like. It also supplies the speed to try next if the user indicates the current speed is not correct by typing a *break* character.

Syntax: *label# initial-flags # final-flags # login-prompt #next-label*

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/group

Pathname: */etc/group*

Description: This file is the definition file for user groups on the system.

Syntax: *groupname*:*epasswd*:*GID*:[*user1,user2*]

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

Dependencies: /etc/passwd

/etc/hosts

Pathname: */etc/hosts*

Description: This file contains information regarding the known hosts on the network.

Syntax: *IP-address hostname alias*[*es*]

MAC Label: dbadmin

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/hosts.equiv

Pathname: */etc/hosts.equiv*

Description: This file contains a list of trusted hosts. When an *rlogin*, *rcp*, or *rsh* request from a listed host is made, and the initiator of the request is also listed in the */etc/passwd* file, no further validity checking is done as long as the login name and user ID number of the user on the remote host are identical to the listing in the local */etc/passwd* file. If these conditions are met, *rlogin* does not prompt for a password, and *rcp*, and *rsh* complete successfully. So a remote user is "equivalenced" to a local user with the same user name and user ID number when the remote user's host name is found in *hosts.equiv*.

Syntax: *hostname*

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/ioctl.syscon

Pathname: */etc/ioctl.syscon*

Description: This file defines the state of the console device. When *init* comes up at boot time, and whenever it switches out of single-user state to normal run states, it sets the *ioctl* states of the virtual console, */dev/console*, to those modes saved in the file */etc/ioctl.syscon*. This file is written by *init* whenever the single-user state is entered.

Syntax: `d26:1805:8bf:3b:0:3:1c:8:18:4:0:0:0:0:0:0`

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

Referenced by: init

Modified by: init

/etc/inittab

Pathname: */etc/inittab*

Description: This file supplies the script to *init*'s role as a general process dispatcher. The majority of *init*'s process dispatching activity involves creating instances of the terminal line process, */etc/getty*. Other processes typically dispatched by *init* are daemons and shells.

Syntax: id:rstate:action:process

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/motd

Pathname: */etc/motd*

Description: This file is used for the "Message of the Day." The system administrator can freely edit this file. The */etc/motd* file is displayed each time a user logs in.

Syntax: ASCII text file.

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/nologin

Pathname: */etc/nologin*

Description: If the file is present, remote user logins via the network are not permitted.

Syntax: There is no syntax to this file. The existence of the file is all that is required.

MAC Label: dblow

DAC Permission:
-r--r--r-- (0444) root,sys

Dependencies: login

Referenced by: login

/etc/opasswd

Pathname: */etc/opasswd*

Description: This file is a backup copy of */etc/passwd*.

Syntax: *username*:*e_passwd*[,*Mmww* | *lock_char*]:*UID*:*GID*:*GECOS*:$*HOME*:$*SHELL*

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/passwd

Pathname: */etc/passwd*

Description: This file contains information about the user. Unlike standard IRIX, the encrypted password is not stored in this file. The encrypted password is kept in */etc/shadow*. The *passwd* file contains the following information for each user:

name	User's login name contains no uppercase characters and must not be greater than eight characters long.
unused	The field that is normally occupied by the password is unused.
numerical user ID	This is the user's ID in the system and it must be unique.
numerical group ID	This is the number of the group that the user belongs to.
user's real name	In some versions of UNIX, this field also contains the user's office location, extension, home phone, and so on.
initial working directory	The directory that the user is in at login. This is known as the ''home'' directory.
shell	The program to use as the command interpreter ("shell") when the user logs in. If the shell field is empty, the Bourne shell (*/bin/sh*) is assumed.

Syntax: *username*::*UID*:*GID*:*GECOS*: $*HOME*:$*SHELL*

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/profile

Pathname: */etc/profile*

Description: This file is the prototype shell environment command file for use with */bin/sh*. Commands in this file are executed when the shell starts up.

Syntax: ASCII text file.

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/rhost.conf

Pathname: */etc/rhost.conf*

Description: This file is the configuration file for the remote login and remote shell programs. It specifies the parameters under which remote logins and shells are allowed on your system from systems that share your security policy and those that do not. Default capability sets and allowed login labels are specified here.

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/services

Pathname: */etc/services*

Description: The */etc/services* file contains information regarding the known services available in the Internet.

Syntax: Example syntax:

```
smtp 25/tcp mail
```

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/shadow

Pathname: */etc/shadow*

Description: This is the user password file. This file contains the following information for each user:

name	User's login name—contains no uppercase characters and must not be longer than eight characters.
password	Encrypted password and optional password aging information

Syntax: The following is a sample *shadow* file:

```
root:kEXFeXFTPoxE
bill:6k/7KCFRPNVXg,z/
```

MAC Label: dbadmin

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/syslog.conf

Pathname: */etc/syslog.conf*

Description: This file directs the system log daemon (*syslogd*) to log messages in a given set of files. Each log message in a logfile is one line. For more information about this file, see the *syslogd*(1m) reference page.

Syntax: An example *syslog.conf* file:

```
kern.debug |/usr/adm/klogpp /usr/adm/SYSLOG
kern.debug |/usr/adm/klogpp /dev/console
daemon,auth,syslog,lpr.debug /usr/adm/SYSLOG
kern.err @ginger
*.emerg *
*.alert eric,beth
*.alert;auth.warning ralph
```

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/ttytype

Pathname: */etc/ttytype*

Description: This file contains a list of the tty ports on the system, and for each port, the kind of terminal that is attached to it.

Syntax: Example:

```
iris-ansi console
iris-ansi systty
vt100 ttyd1
?h19 ttyd2
?h19 ttyd3
?v50am ttyd4
?v50am ttyd5
?v50am ttyd6
?v50am ttyd7
?v50am ttyd8
?v50am ttyd9
?v50am ttyd10
?v50am ttyd11
?v50am ttyd12
```

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/utmp

Pathname: */etc/utmp*

Description: This file holds user information for such commands as *who*, *write*, and *login*. For more information about this file, see the reference page *utmp*(4).

Syntax: Example:

```
struct utmp {
char ut_user[8]; /*User login name*/
char ut_id[4]; /*/etc/inittab id usually line #)*/
char ut_line[12]; /* device name (console,lnxx)*/
short ut_pid; /*process id*/
short ut_type; /* type of entry */
struct exit_status {
~~~~short ~~~~e_termination;/*termination status*/
~~~~short ~~~~e_exit; /* Process exit status */
}ut_exit; /*exit status of a process marked */
/* as a DEAD_PROCESS.*/
time_t ut_time; /* time entry was made */
};
```

MAC Label: dblow

DAC Permission:

-rw-rw-r-- (664) root,sys

/etc/wtmp

Pathname: */etc/wtmp*

Description: This file contains one record per username with related information: inittab ID, device name connected to, process ID, type of entry (for example, a login process), exit status, and time the entry was made. For more information about this file, see the reference page *utmp*(4).

Syntax: Example:

```
struct utmp {
char ut_user[8]; /* User login name */
char ut_id[4]; /*/etc/inittab id usually line #*/
char ut_line[12]; /* device name (console,lnxx) */
short ut_pid; /* process id */
short ut_type; /* type of entry */
struct exit_status {
~~~~short ~~~~e_termination;/*termination status*/
~~~~short ~~~~e_exit; /* Process exit status */
} ut_exit; /* The exit status of a process marked as
DEAD_PROCESS. */
time_t ut_time; /* time entry was made */
};
```

MAC Label: dblow

DAC Permission:
-rw-rw-r-- (664) root,sys

Files in the /etc/config Directory Structure

All files in the config directory that lack suffixes contain only the words ‘‘on’’ or ‘‘off.’’ This indicates whether or not the named subsystem is activated at system startup time. Files with the suffix ".options" contain flags to the subsystem startup command.

/etc/config/acct

Pathname: */etc/config/acct*

Description: This file contains either the word "on" or "off." If it contains "on," process accounting is turned on by default. If it contains the word "off," process accounting is not run by default.

Syntax: The word "on" or "off."

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/config/automount

Pathname: */etc/config/automount*

Description: This file is used by the system to direct NFS to automatically mount network filesystems or not to mount them.

Syntax: The word "on" or "off."

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/config/login.options

Pathname: */etc/config/login.options*

Description: This file controls the default actions of the *login* program, such as the number of unsuccessful attempts to log in or the timeout period while waiting for a password. This file is described in the *login*(4) reference page.

Syntax: Example:

```
maxtries=5
disabletime=30
passwdreq
```

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/config/named

Pathname: */etc/config/named*

Description: This file directs the system to spawn or not to spawn the *named* domain name server.

Syntax: The word “on” or “off.”

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/config/network

Pathname: */etc/config/network*

Description: This file is used by the system to direct NFS to spawn the lock and status daemons or not to spawn them.

Syntax: The word “on” or “off.”

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/config/nfs

Pathname: */etc/config/nfs*

Description: This file is used by the system to start the NFS daemons and mount the network filesystems.

Syntax: The word “on” or “off.”

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

Referenced by: init

/etc/config/rwhod

Pathname: */etc/config/rwhod*

Description: This file directs the system to spawn or not to spawn the *rwhod* server daemon.

Syntax: The word “on” or “off.”

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/config/satd.options

Pathname: */etc/config/satd.options*

Description: This file contains saved *satd* options. A flag to *satd* fills this file with the current *satd* options.

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/config/sat_select.options

Pathname: */etc/config/sat_select.options*

Description: This file contains saved options to *sat_select*. A flag to *sat_select* fills this file with the current *sat_select* options.

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/config/syslogd.options

Pathname: */etc/config/syslogd.options*

Description: This file contains command line options for the *syslogd* program. *syslogd* reads and logs messages into a set of files. For information about the *syslogd* program, see the syslogd(1M) reference page.

Syntax: Optional site-specific flags belong in the options file. The available flags are:

-f – Specify an alternate configuration file.

-m – Select the number of minutes between mark messages.

-d – Turn on debugging.

-p – Use the given name for the named pipe instead of */dev/log*.

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

/etc/config/timed

Pathname: */etc/config/timed*

Description: This file directs the system to spawn or not to spawn the *timed* clock controlling daemon. For more information about *timed*, see the timed(1M) reference page.

Syntax: The word “on” or “off.”

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

Fields in the /etc/mac File Structure

The */etc/mac* file structure contains seven fields: categorynames, devisionnames, labelnames, lbldb_bin, levelnames, minttypenames, and msentypenames. The */etc/mac* file has the following label and permission:

MAC Label: dblow

DAC Permission:
-rw-r--r-- (644) root,sys

categorynames

Description: This field associates a human readable text name with the category number. Typically, the category number is used to identify projects or areas (categories) of information. A category allows labels to have the same sensitivity level (such as “proprietary”) but different security labels by having different categories (such as a category named “cashew” and a category named “pistachio”).

Syntax: A *categorynames* field might look like:

```
black:10
blue:11
green:63
lavender:110
orange:140
pink:150
purple:151
red:170
violet:210
white:220
```

divisionnames

Description: This field associates a human readable text name with the division number. The division number is a non-hierarchical value that indicates a grade type that this label includes. Typically, the division number is used to identify projects or areas (categories) of integrity. A division allows labels to have the same grade (such as "highest grade"), and to have different security labels by having different divisions (such as a division named "huguenots" and a division named "papists").

Syntax: A *divisionnames* field might look like this:

```
IRIXdbadmin:1
IRIXsuperuser:2
IRIXinit:3
“apple pie”:101
cake:120
chocolate:121
cookie:122
custard:124
fudge:150
```

gradenames

Description: This field associates a human readable ASCII text name with the grade value. The grade value is a hierarchical value that indicates how trustworthy the label is. Typically, the grade has names such as Dubious, Suspect, Normal, Confident, Verified, TheTruth, and so on.

Syntax: A *gradenames* field might look like this:

```
lowestgrade:0
“lowest grade”:0
good:0
choice:10
prime:80
best:255
highestgrade:255
“highest grade”:255
```

labelnames

Description: This field defines all the possible security labels in an ASCII text format. It associates an ASCII text name with a security label. This field defines the name of the security label, and the MSEN type, sensitivity level, MINT type, grade, categories and divisions. Interpretation of this field relies on the *msentypenames*, *levelnames*, *minttypenames*, *gradenames*, *categorynames*, and *divisionnames* data files to define components that are specified by name rather than by number.

Syntax: A *labelnames* field might look like this:

```
label1:"msenequal/mintlow"
label2:"msenlow/mintlow"
label3:"tcsec,public/mint=biba,grade=good"
```

lbldb_bin

Description: This field is the system binary label database. All system labels are listed here in binary form.

Syntax Binary file.

levelnames

Description: This field associates a human readable ASCII text name with the sensitivity level value. The sensitivity level is a hierarchical value that indicates how "sensitive" the label is. Typically, the sensitivity level has names such as Sensitive, Classified, Secret, Top Secret, and so on.

Syntax: A typical *levelnames* field looks something like this:

```
# This is a comment line.
public:0
proprietary:30
"company sensitive":40
"company confidential":50
"executive committee only":60
```

minttypenames

Description: This field maps the MINT (Mandatory INTegrity) type of a security label from an ASCII name to the numeric value.

Syntax: The default *minttypenames* field on your system looks like this:

```
MintEqual:0x65
MintHigh:0x68
MintLow:0x6c
Biba:0x62
```

msentypenames

Description: This field maps the MSEN type of a security label from an ASCII name to the numeric value.

Syntax: The default *msentypenames* field on your system looks like:

```
Admin:0x41
MsenEqual:0x45
MsenHigh:0x48
MldHigh:0x49
MsenLow:0x4c
Mld:0x4d
MldLow:0x4e
tcsec:0x54
Tcsec:0x54
```

Files in the /usr Directory Structure

/usr/adm/lastlog/username

Pathname: */usr/adm/lastlog/username*

Description: These files record information for use by the *login* program about your last login.

Syntax: A typical *lastlog* file might look like:

```
^A(:4ujohnsmith.other.place.com
```

MAC Label: dbadmin

DAC Permission:
-rwxr-xr-x (755) root,sys

/usr/adm/oSYSLOG

Pathname: */usr/adm/oSYSLOG*

Description: This file is a saved old version of the system log.

Syntax: A typical *oSYSLOG* has records of the form:

```
Sep 2 01:01:38 mymachine syslogd: restart
Sep 3 15:26:12 mymachine sendmail[15324]: AA15324:
from=, size=1027, class=0
Sep 3 17:14:02 mymachine sendmail[15424]: AA15424:
from=, size=1080, class=0
Sep 3 17:44:03 mymachine sendmail[15461]: AA15461:
from=, size=974, class=0
```

MAC Label: dbadmin

DAC Permission:
-rw-r--r-- (644) root,sys

/usr/adm/SYSLOG

Pathname: */usr/adm/SYSLOG*

Description: This file contains a log of all events corresponding to those selected in the */etc/syslog.conf* file.

Syntax: A typical *SYSLOG* file looks like:

```
Sep 2 01:01:39 mymachine syslogd: restart
Sep 3 09:58:35 mymachine sendmail[21326]: AA21326:
from=, size=2266, class=0
Sep 3 10:02:32 mymachine sendmail[21336]: AA21336:
from=, size=1605, class=0
Sep 3 10:07:15 mymachine sendmail[21342]: AA21342:
from=, size=2202, class=0
```

MAC Label: dbadmin

DAC Permission:
-rw-r--r-- (644) root,sys

/usr/lib/X11/xdm/Xresources

Pathname: */usr/lib/X11/xdm/Xresources*

Description: This file contains default information about your X environment.

Syntax: The default *Xresources* file looks like this:

```
xlogin*login.translations: #override
<key> F1: set-session-argument(failsafe) finish-field()
<key> Return: set-session-argument() finish-field()
xlogin*borderWidth: 3
#ifdef COLOR
xlogin*greetColor: #f63
xlogin*failColor: red
xlogin*Foreground: black
xlogin*Background: #fdc
#else
xlogin*Foreground: black
xlogin*Background: white
#endif
```

MAC Label: dblow

DAC Permission:
-r--r--r-- (0444) root,sys

/usr/spool/lp/pstatus

Pathname: */usr/spool/lp/pstatus*

Description: Printer status information is stored in this file.

Syntax: Data file.

MAC Label: dbadmin

DAC Permission:
-rw-r--r-- (644) lp,sys

/usr/spool/lp/qstatus

Pathname: */usr/spool/lp/qstatus*

Description: Print queue status information is stored in this file.

Syntax: Data file.

MAC Label: dbadmin

DAC Permission:
-rw-r--r-- (644) lp,sys

Chapter 9

Administering Printing and Tape Devices

This chapter assumes that you are familiar with the standard IRIX treatment of printing and tape devices, including system backup and restoration strategies. Complete information on the use of peripheral devices with Silicon Graphics systems can be found in the guide titled *IRIX Admin: Peripheral Devices*. Further, complete information on system backups can be found in the guide titled *IRIX Admin: Backup, Security, and Accounting*.

Sections in this chapter include:

Printing Under Trusted IRIX/CMW

Printing under the Trusted IRIX/CMW system requires no special resources. Except where noted in this chapter, printing operates exactly as described in your standard IRIX documentation. Trusted IRIX/CMW meets the requirement for B1-level systems for labeled printing. Each page of printed output carries the label of the file printed 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. Line printing under Trusted IRIX/CMW is essentially the same as under standard IRIX, except that printed copy is labeled and fewer printer options are supported. The printer daemon process (see the *lpsched*(8) reference page), must be run from the system startup scripts. It can be stopped and restarted while the system is running with the following commands:

```
/etc/init.d/lp stop
/etc/init.d/lp start
```

Supported Printers

Trusted IRIX/CMW supports line printing on ASCII (dumb) printers and PostScript printers. The utilities that allow labeled PostScript output, however, are not resistant to label spoofing programs. Because of this weakness, it is up to the individual system administrator to determine whether PostScript printing can be allowed at the site.

Labeling Printer Output

This section defines the methods implemented to properly label printed output. There are several parts to the printing system: the print job submission program, the program that produces the output (in the case of PostScript), and the program that labels the output.

PostScript Printer Output

PostScript requires a print job to be written in the PostScript definition language. This language specifies the parameters and specifics of the printout. Trusted IRIX/CMW has implemented a filter to this output production program that attaches labels to the individual page specifications and creates an appropriately labeled banner page.

ASCII Printer Output

ASCII printers rely on escape sequences within the print stream to provide directions to the printer. Trusted IRIX/CMW has implemented a filter for ASCII print jobs that inserts the labels into the print stream.

Printing Software

Trusted IRIX/CMW has implemented the *lp* and *pr* utilities to produce labeled printer output. Using the information supplied here, the system administrator can extend support to other printers. Printing interface utilities under UNIX are usually in the form of shell scripts that are invoked by the *lp* command. The usual MAC policies are implemented around printing requests. The print request inherits the label of the file being printed, and this label is used to control access to the print job. For example, MAC must be satisfied in order to cancel the print job or to call up the job on the printer spooler queue. When printing on an ASCII printer, the print job is sent through the *pr* filter program with the **-b** option in order to filter out escape sequences and apply the internal page labels.

Other optional arguments to the *pr* program are **-l** and **-f**, followed by the filename and the name of the type of printer.

Configuring a Parallel Printer under Trusted IRIX/CMW

The following procedure configures the device */dev/plp* as a dumb printer named "elephant."

Log in as root at *dblow* and enter the following commands in order:

1. This command stops the printing spooler while the operation takes place.
   ```
   /etc/init.d/lp stop
   ```
2. This command changes your label to *dbadmin* to perform the operation.
   ```
   newlabel -F dbadmin
   ```
3. This command directs *lpadmin* to create the dumb printer "elephant" on */dev/plp*.
   ```
   /usr/lib/lpadmin -pelephant -mdumb -v/dev/plp
   ```
4. This command changes your label back to *dblow* to restart the printing spooler.
   ```
   newlabel -F dblow /etc/init.d/lp start
   ```
5. This command enables the new printer.
   ```
   /usr/bin/enable elephant
   ```
6. This command directs the new printer to begin accepting requests.
   ```
   /usr/lib/accept elephant
   ```
7. This command directs *lpadmin* to make the new printer the default printer.
   ```
   /usr/lib/lpadmin -delephant
   ```
8. This command sets the MAC label range that the printer will accept for jobs.
   ```
   /usr/lib/lpadmin -s msenhigh/mintlow...msenlow/minthigh -pelephant
   ```
9. This command confirms that the printer is enabled and accepting requests.
   ```
   /usr/bin/lpstat -a
   ```

This process initializes */dev/plp* and */dev/plpbi* to the label *dbadmin*. Some larger systems, such as CHALLENGE and Onyx systems, have multiple */dev/plp* ports and if you are installing printers on these ports, be sure that each one has been labeled at *dbadmin*..

If you are installing a serial printer, you can use any */dev/ttyd** port, and that port must be labeled at *dbadmin*.

Magnetic Tape Backups

One of the most important responsibilities of the system administrator is that of preventive maintenance. It is very important to create frequent backups of all files on the system. It is far less painful to recover a system whose files are a day old than it is to start from scratch. If you back up your entire filesystem at least weekly and back up changed files every day, you can maintain a reasonable assurance that the data contained on your backups is uncorrupted and current.

The original distribution media for your system should always be stored in a safe place.

After your trusted software is installed and configured, but before you allow users to begin work, make a complete backup of your system using *tar* and make a record of all your system files, their attributes and a checksum, and store this backup with your distribution media. With this record and the original tapes you should be able to recreate your system if needed.

Backups should be done by the individual users in a workstation environment or by the system administrator if a server is used. The specific backup practices at any given site should be approved by the system administrator. The tape device for the Trusted IRIX/CMW system (*/dev/tape*) is shipped with an exact label. The system administrator must change this label each time a user at a different label wishes to use the tape device.

tar Backups Under Trusted IRIX/CMW

B1 systems are required to provide for labeled tape backups. 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 on all files placed on the tape. To recover files from backup, use *tar* with the **M** keyword. Always remember that it is still possible to make unlabeled backups using *tar* without the **M** keyword. Also, using *tar* to extract labeled files without the **M** keyword results in the loss of label data. It is therefore strongly recommended that access to the physical tape device and possession of magnetic tapes be limited to the system administrator. Even though *tar* maintains labeling on the tape, the act of making a tape is still subject to MAC. Assuming that *root* makes the system backups, *root* should follow this procedure for system backups:

1. Make sure that *root* has read privilege to all directories and files.
2. Use the *chlabel* command to change the label of the tape device to match your label.
3. Change directories to the directory you wish to back up.

4. Enter the following command to begin the backup:

   ```
   tar cvM .
   ```

5. Write the highest label on your system on the surface of the tape cartridge, so it is not inadvertently made available or discarded.

Recovering files in this manner is the reverse of removal. You must make certain that the tape device is properly labeled and then you can restore files using the tape you made previously. If all the files in a directory are known to be at a single label, you can log in with sufficient clearance and change the label of the tape device to match the directory label and make a single level backup. You should still use the **M** keyword to *tar*, however, to maintain the label information. Also, write the label of the information on the tape on the surface of the tape cartridge.

Remote Tape Drives

A program called */etc/rmt* in the Trusted IRIX/CMW system allows you to use the remote tape drive feature of *tar*. The */etc/rmt* file is distributed with the label binary.

To use the remote tape drive features of *tar* over a Monolabel network connection, you must change the label of */etc/rmt* to match the label of your Monolabel network.

Chapter 10

Maintaining an Evaluated Configuration

If you intend to run the evaluated configuration of the Trusted IRIX/CMWsystem at your site, you should be aware that there are strict limits placed on the hardware and software that have been evaluated. This chapter defines the evaluated configuration. If anything is added to, modified in, or subtracted from the evaluated configuration as described in this chapter, you are no longer running the evaluated configuration. Your system may operate normally, but the configuration has not been tested for security.

This chapter covers these topics:

Hardware Configuration

The Trusted IRIX/CMW system is under evaluation, and the evaluated hardware list has not been finalized at the time of publication. However, Trusted IRIX/CMW runs on the entire Silicon Graphics family of workstations and servers.

The following peripheral devices are being evaluated with Trusted IRIX/CMW:

Disk, magnetic tape, CD

Serial (dumb) printers
: As described in "Supported Printers" in Chapter 9, ''Administering Printing and Tape Devices,'' serial (dumb) ASCII printers are supported under the evaluated configuration of Trusted IRIX/CMW.

Network connections
: As described in Chapter 4, "Networking With Trusted IRIX/CMW," the standard network connections supplied are supported.

Software Configuration

The Trusted IRIX/CMW software must be kept intact and used as directed to maintain the evaluated configuration.

Use of the minthigh Integrity Label

Only software that is part of the evaluated Trusted Computing Base (TCB) shipped with Trusted IRIX/CMW may use the *minthigh* integrity label. This level of integrity requires formal security evaluation. If you change the integrity grade of any other file to *minthigh*, you are no longer running the evaluated configuration. Label names with *minthigh* are *dblow* and *dbadmin*.

TCB Files and Programs

The files and programs that make up the Trusted IRIX/CMW TCB can be derived from the */etc/irix.mac* directory. All files labeled *dblow* and *dbadmin* are part of the TCB.

Administrative Configuration

The administrative settings shipped with Trusted IRIX/CMW are part of the evaluated configuration.

Login Options

Trusted IRIX/CMW is shipped with a default set of login options set in the */etc/config/login.options* file. If you change any of these options, you are no longer running the evaluated configuration.

Networking

As described in Chapter 4, “Networking With Trusted IRIX/CMW,” the evaluated configuration requires that you run only the *tsix* network software on all network interfaces. Any additional network connections violate the evaluated configuration.

Filesystems

The following sections concern filesystem configuration under Trusted IRIX/CMW.

- Labeled Filesystems
- NFS Exports

Labeled Filesystems

All filesystems under Trusted IRIX/CMW must be labeled at all times. There are no exceptions to this rule. Even filesystems on nontrusted systems are assigned a label when connected via the network to Trusted IRIX/CMW systems.

NFS Exports

All NFS exported filesystems must be exported using the XFS extended attribute NFS extension, preserving the classifications of files across the Trusted IRIX/CMW network.

For more information, see Chapter 4, “Networking With Trusted IRIX/CMW.”

Printers

The evaluated configuration of Trusted IRIX/CMW supports printing on dumb serial printers only. For complete information, see “Printing Under Trusted IRIX/CMW” on page 156.

Index

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

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 007-3299-002.

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