Guide to VMS System Security

Order Number: AA-LA40A-TE

April 1988

This guide describes the security features available through the VMS operating system. It explains the purpose and proper application of each feature in the context of specific security needs.

Revision/Update Information: This document supersedes the *Guide to VAX/VMS System Security*, Version 4.4.

Software Version: VMS Version 5.0

digital equipment corporation
maynard, massachusetts

April 1988

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem–10	PDP	VT
DECSYSTEM–20	PDT	
DECUS	RSTS	
DECwriter	RSX	digital™

ZK4525

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

USA & PUERTO RICO*

Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire
03061

CANADA

Digital Equipment
of Canada Ltd.
100 Herzberg Road
Kanata, Ontario K2K 2A6
Attn: Direct Order Desk

INTERNATIONAL

Digital Equipment Corporation
PSG Business Manager
c/o Digital's local subsidiary
or approved distributor

In Continental USA and Puerto Rico call 800-258-1710.

In New Hampshire, Alaska, and Hawaii call 603-884-6660.

In Canada call 800-267-6215.

*Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575).

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminster, Massachusetts 01473.

Production Note

This book was produced with the VAX DOCUMENT electronic publishing system, a software tool developed and sold by DIGITAL. In this system, writers use an ASCII text editor to create source files containing text and English-like code; this code labels the structural elements of the document, such as chapters, paragraphs, and tables. The VAX DOCUMENT software, which runs on the VMS operating system, interprets the code to format the text, generate a table of contents and index, and paginate the entire document. Writers can print the document on the terminal or line printer, or they can use DIGITAL-supported devices, such as the LN03 laser printer and PostScript™ printers (PrintServer 40 or LN03R ScriptPrinter), to produce a typeset-quality copy containing integrated graphics.

™ PostScript is a trademark of Adobe Systems, Inc.

Contents

EXAMPLES

FIGURES

TABLES

Preface

Intended Audience

This guide is designed for VMS users responsible for protecting operating systems from tampering, observation, or theft of services by unauthorized users. The term *security manager* is used in this guide to refer to the person or persons responsible for system security.

Document Structure

This guide consists of the following:

- Chapter 1 discusses levels of security requirements and describes three sources of security failures.
- Chapter 2 introduces the reference monitor concept of security design and provides an overview of VMS security features.
- Chapter 3 provides general information about system security.
- Chapter 4 describes VMS file protection features in detail.
- Chapter 5 describes general system security features.
- Chapter 6 describes how to recognize when a system is under attack and protective/defensive actions available.
- Chapter 7 describes security considerations for systems using networking.
- Chapter 8 describes security-related actions specific to clustered systems, such as mounting the disks and setting up the authorization database.
- Appendix A summarizes all user privileges available on VMS systems, what they provide, and who may need them.
- Appendix B describes how to access the user data areas in the User Authorization File.
- Appendix C lists the default UIC-based protection that DIGITAL provides for system files.
- Appendix D describes how to operate VMS systems in a C2 environment.
- Appendix E lists alarm messages.
- The Glossary offers definitions of security-related terms introduced in this guide.

Associated Documents

To effectively implement VMS security features, you should be familiar with the system management information presented in the following manuals:

- *Introduction to VMS System Management*
- *Guide to Setting Up a VMS System*
- *Guide to Maintaining a VMS System*

Users with a specific interest, such as in networking or clusters, should be familiar with the documents provided in the documentation set for their areas. Security managers in VMS networking environments should be familiar with the *VMS Networking Manual*. Security managers on clustered systems should be familiar with the *VMS VAXcluster Manual*.

Conventions

Convention	Meaning
RET	In examples, a key name (usually abbreviated) shown within a box indicates that you press a key on the keyboard; in text, a key name is not enclosed in a box. In this example, the key is the RETURN key. (Note that the RETURN key is not usually shown in syntax statements or in all examples; however, assume that you must press the RETURN key after entering a command or responding to a prompt.)
CTRL/C	A key combination, shown in uppercase with a slash separating two key names, indicates that you hold down the first key while you press the second key. For example, the key combination CTRL/C indicates that you hold down the key labeled CTRL while you press the key labeled C. In examples, a key combination is enclosed in a box.
$ SHOW TIME 05-JUN-1988 11:55:22	In examples, system output (what the system displays) is shown in black. User input (what you enter) is shown in red.
$ TYPE MYFILE.DAT . . .	In examples, a vertical series of periods, or ellipsis, means either that not all the data that the system would display in response to a command is shown or that not all the data a user would enter is shown.
input-file, . . .	In examples, a horizontal ellipsis indicates that additional parameters, values, or other information can be entered, that preceding items can be repeated one or more times, or that optional arguments in a statement have been omitted.

Convention	Meaning
[logical-name]	Brackets indicate that the enclosed item is optional. (Brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks ("). The term apostrophe (') is used to refer to a single quotation mark.

New and Changed Features

Version 5.0 of VMS contains the following new security features:

- Access control lists (ACLs) on queues
- Enhanced proxy login features
- Forcing expired password change
- True highwater marking for sequential, exclusively-accessed files

1 Introduction

Effective operating system security measures help prevent unauthorized access and theft of proprietary software, software plans, and computer time. These measures can also protect equipment, software, and files from damage caused by tampering.

This chapter provides security managers with an overview of security measures available with the VMS operating system.

1.1 Types of Computer Security Problems

The source of a security breach on a computer system can usually be traced to one of three categories: user irresponsibility, user probing, or user penetration.

1.1.1 User Irresponsibility

User irresponsibility refers to situations where the user purposely or accidentally causes some noticeable damage. An example would be a user who is authorized to access certain files making a copy of a key file to sell.

There is little that an operating system can do to protect sites from this source of security failures. The problem frequently lies in application design deficiencies or inconsistent use of available controls by users and the security manager. Sometimes the failure to enforce adequate environmental security unwittingly encourages this type of security problem.

Even the best security system will fail if implemented inconsistently. This, along with the failure to motivate your users to observe good security practices, will make your system vulnerable to security failures caused by user irresponsibility.

1.1.2 User Probing

User probing refers to situations where a user exploits insufficiently protected parts of the system. Some users consider gaining access to a forbidden system area as an intellectual challenge, playing a game of user-versus-system. Although intentions may be harmless, theft of services is a crime. Users with more serious intent may seek confidential information, attempt embezzlement, or even destroy data by probing. Always treat user probing seriously.

VMS provides many security features to combat user probing. Based on security needs, the security manager implements features either on a temporary or permanent basis. These features are discussed in later sections.

1.1.3 User Penetration

Penetration refers to situations where the user breaks through security controls to gain access to the system. While VMS has security features making penetration extremely difficult, it is impossible to make any operating system completely impenetrable.

A user who succeeds in penetrating a system is both skilled and malicious. Thus, penetration is the most serious and potentially dangerous type of security breach. With proper implementation of VMS security features, however, it is also the rarest security breach, requiring unusual skills and perseverance.

1.2 Levels of Security Requirements

Each site has unique security requirements. Some sites may need limited measures because they are able to tolerate some forms of unauthorized access with little adverse effect. At the other extreme are those sites that cannot tolerate even the slightest probing, such as strategic military defense centers. In between are many commercial sites, such as banks.

To ascertain your security requirements, answer the questions in Table 1–1. Your answers may help determine your security needs to be low, medium, or high.

Table 1–1 Event Tolerance as a Measure of Security Requirements

Question: Could You Tolerate the Following Event?	Level of Security Requirements Based on Toleration Responses		
	Low	Medium	High
A user knowing the images being executed on your system	Y	Y	N
A user knowing the names of another user's files	Y	Y	N
A user accessing the file of another user in the group	Y	Y	N
An outsider knowing the name of the system just dialed into	Y	Y	N
A user copying files of other users	Y	N	N
A user reading another user's electronic mail	Y	N	N
A user writing data into another user's file	Y	N	N
A user deleting another user's file	Y	N	N
A user being able to read sections of a disk that might contain various old files	Y	N	N

Table 1–1 (Cont.) Event Tolerance as a Measure of Security Requirements

Question: Could You Tolerate the Following Event?	Level of Security Requirements Based on Toleration Responses		
	Low	Medium	High
A user consuming machine time and resources to perform unrelated or unauthorized work, possibly even playing games	Y	N	N

If you can tolerate most of the events listed, your security requirements are quite low. If your answers are equally mixed between yes and no, your requirements are in the medium to high range. Generally, those sites that are most intolerant to the events have very high levels of security requirements.

When reviewing security needs, do not confuse a weakness in site operations or recovery procedures as a security problem. Ensure that your operations policies are effective and consistent before evaluating your system security requirements.

1.3 The Secure System Environment

There are two sources of security problems outside the operating system domain: employee carelessness and facility vulnerability. If you have a careless or malicious employee or your facility is insecure, none of the security measures discussed in this guide will protect you from security breaches.

Most system penetration occurs through these environmental weaknesses. It is much easier to physically remove a small reel of tape than it is to break access protection codes or change file protection.

DIGITAL strongly encourages you to stress environmental considerations as well as operating system protections when reviewing site security.

The following chapters discuss VMS operating system security measures. When deciding on which of these measures to implement, it is important for you to assess site security needs realistically. While instituting adequate security for your site is essential, instituting more security than actually necessary is costly and time-consuming.

When deciding which security measures to apply to your system, remember the following:

- The most secure system is also the most difficult to use.
- Increasing security can increase costs in terms of slower access to data, slower machine operations, and slower system performance.
- More security measures require more personnel time. (Increased security requires increased employee hours.)

The VMS operating system provides the basic mechanisms to control access to the system and its data. It also provides monitoring tools to ensure that access is restricted to authorized users. However, many computer crimes are committed by authorized users with no violation of the operating system's security controls.

Therefore, the security of your operation depends on how you apply these security features and how you control your employees and your site. By first building appropriate supervisory controls into your application and designing your application with the goal of minimizing opportunities for abuse, you can then implement VMS operating system security features and produce a less vulnerable environment.

2 Overview

This chapter presents the concepts that guided the design and implementation of the security features and mechanisms incorporated into the VMS operating system. The intent is to provide a framework for thinking about your total system security picture. Subsequent chapters present details about the security features and their use.

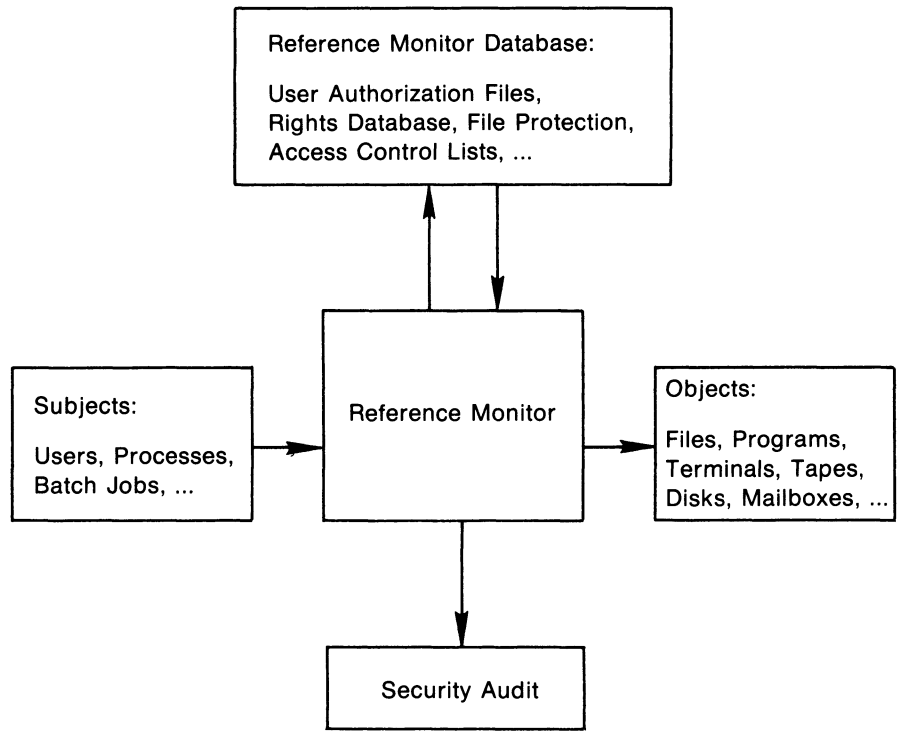
2.1 The Reference Monitor Concept

In the late 1960s, a great deal of research and development was dedicated to the problem of achieving security in multiuser computer systems. Much of the development work involved attempts to find "all the things that could go wrong" with a system's security and then to correct those flaws one by one. It became apparent to the researchers that this process was ineffective; effective system security could only result from a basic model of the structure of a secure computer system. The reference monitor concept was proposed as such a model and gained wide acceptance.

The reference monitor concept depicts a computer system in terms of subjects, objects, an authorization database, an audit trail, and a reference monitor mechanism. Figure 2–1 shows the relationship of these elements. *Subjects* are active entities that gain access to information on behalf of people, such as a user process. *Objects* are passive repositories of information to be protected, such as a file. The *authorization database* defines the system's security requirements by revealing which subjects (acting on behalf of users) can have which kinds of access to objects (that contain information). The *audit trail* maintains a record of access attempts, successful or not, as required by the authorization database.

2.1 The Reference Monitor Concept

Figure 2–1 Reference Monitor Diagram



The reference monitor mechanism enforces the security rules by authorizing the creation of subjects, granting subjects access to objects according to the requirements of the database, and recording events as necessary in the audit trail. In an ideal system, the reference monitor mechanism is required to meet the following three requirements:

- Mediate every attempt by a subject to gain access to an object
- Provide a tamperproof database and audit trail that are thoroughly protected from unauthorized observation and modification
- Remain small, simple, and well-structured so that effectiveness in enforcing security requirements can be assured

A system that fully meets all requirements of the reference monitor model would be very secure. These are the requirements proposed for systems that are secure even against penetration. (In such systems, the reference monitor is implemented by a security-related subset, or *security kernel*, of the operating system.) While VMS is not such a system, its interface to users and system managers does mirror the basic structure dictated by the reference monitor concept. Experience shows that incorporating such a structure is the best way to build a system resistant to probing and to most attempts at penetration.

2.2 The Reference Monitor and VMS

The following sections explain the components of the VMS operating system comparable to the roles of subjects, objects, authorization database, audit trail, and reference monitor mechanism.

2.2.1 Subjects

In VMS, the process performs the role of the subject, which is the active element that gains access to information. When a user logs in to use VMS interactively, or when a batch or network job starts, VMS creates a process that includes the identity of the user. That process gains access to information as the agent for the user.

Processes are vulnerable to security breaches during creation and while accessing information. VMS manages process access to information using its authorization data and internal mechanisms. Process creation has many areas of security vulnerability. For this reason, many of the security features in VMS concentrate on the area of process (or subject) creation.

When a user attempts to log in to VMS, the user provides a *user name* (a name that will be given to the resulting process) and a *password*. The password serves as an authenticator that should be known only to the user and to VMS. Because a short or obvious password is likely to fail this requirement, VMS incorporates many password protection mechanisms that can be invoked by the user or required by the security manager. VMS is also capable of limiting the number of attempts that an intruder can make to guess a password. Briefly, then, the association of the user with a subject (or process) is a critical aspect of system security.

The file of users' passwords is part of the reference monitor's database that must be protected from unauthorized observation and modification. VMS attempts to meet this requirement by storing the password in a file normally protected from general access, the *system user authorization file* (SYSUAF.DAT). VMS also takes the precaution of storing passwords in an encoded form that is not usable if stolen.

Once it creates a process, VMS assigns that process a *User Identification Code*, or *UIC*. The UIC corresponds to the name of the user who created the process (as authenticated by the user's password). In addition, the UIC indicates the user's membership in a group that can correspond to the user's department, project, or function. VMS can also attach additional information to the process regarding the creation of the process and the affiliation of the process owner with other groups. This additional information plays a part in the application of the authorization database, which is described in a later section.

2.2.2 Objects

In the reference monitor concept, objects are passive repositories of information. In VMS, there are many objects subject to protection. The most basic (and usually most important) objects in a VMS system are files and directories. The VMS operating system protects files and directories from unauthorized access and provides a variety of mechanisms (outlined in the description of the authorization database) for their controlled sharing.

Objects other than files and directories can also be used to store sensitive information. These objects include sections, mailboxes, logical names, event flag clusters, and queues. VMS provides mechanisms to protect these objects as well, although these protection techniques may not be as complete as those that apply to files.

2.2.3 Authorization Database

In the reference monitor concept, each subject's authorization to gain access to an object is found in a single authorization database. In VMS, the database is distributed and stored in association with the objects that must be protected. For example, the authorization data for a file or directory is stored in the file header for that file or directory.

As the discussion of objects suggested, different objects in VMS can be shared with differing levels of flexibility. Almost all objects are subject to a *UIC-based protection.* This form of protection specifies whether access is allowed or denied to processes running on behalf of the system management, the user who is owner of the object, other members of the UIC group of the owner, and all other users.

In addition to the UIC-based protection, many objects also can be shared under control of *access control lists (ACLs).* ACLs list individual users or groups of users who are to be allowed or denied access to the file or directory. ACLs specify sharing on the basis of UIC as well as other groupings or *identifiers* that can be associated with a process. For example, it is possible to specify that a file should never be read by a process connected to a terminal on a dialup line. Section 4.3 describes ACLs and identifiers in detail.

2.2.4 Audit Trail

A terminal can be designated as a security alarm console where all auditable events are displayed. Some events, such as certain login failures, are always auditable. Other events, such as successful or unsuccessful attempts to gain access to sensitive files, can be selected by users or security managers for auditing. For example, the owner of a sensitive file might create an ACL entry requesting that all accesses to that file be audited.

The audit trail mechanism allows users and security managers to record many events. Because it is time-consuming to examine every event, it is most efficient only to audit events that will contribute the most information to your security picture.

2.2.5 Reference Monitor Mechanism

The VMS executive performs the role of the reference monitor mechanism. All system programs that run in kernel and executive mode help implement the reference monitor, as do certain user-mode images that run with privilege. While the volume of code comprising the VMS executive is large, DIGITAL attempts to ensure that none of the code can be used to bypass system security.

Some VMS privileges can grant a user the authority to modify or subvert the reference monitor mechanism. For example, a process with the BYPASS privilege can gain access to any object without reference to the authorization database. Clearly, the granting of such critical privileges should be severely limited.

Similarly, such privileges as SYSPRV and SECURITY are given to users whose processes help maintain the reference monitor mechanism and database.

2.3 Summary

When designing an overall system security plan, the following questions are pertinent:

- How are users associated with subjects? What is the reliability of the authentication mechanism?
- What objects contain sensitive information in this system or application? Is access to those objects controlled?
- Does the authorization database reflect policy? Who is authorized to gain access to sensitive objects? Are adequate restrictions in place?
- Is the audit trail recording enough or too much information? Who will monitor it?
- What programs are functioning as part of the reference monitor mechanism? Which users can modify the mechanism and the authorization database? Is this the desired configuration?

These considerations, as well as the underlying reference monitor concept, apply equally to a time-shared VMS system, a widespread network, or a single application on a VMS system that grants access to records in a file or database. The VMS operating system provides general mechanisms that users and system managers must apply to achieve system security.

3 Security for the User

This chapter provides overall information about security features of the VMS operating system and is of interest to all users. By reading this chapter and becoming familiar with the additional references for the topics provided throughout, the general user should acquire all the basic information needed to use the system securely. Whether or not users apply this knowledge consistently and accurately while observing the site's specific security policies will make a great difference between a secure system and one that is vulnerable.

3.1 Logging In to the System

The actions you perform to gain access to the system are known as logging in. The system checks user privilege for the first time during the login procedure. During login, users must identify themselves with a user name and password. Depending on site security requirements, the login procedure can be made more elaborate and can increase user restrictions.

3.1.1 Types of Logins

VMS recognizes the following seven types of logins:

- Local
- Dialup
- Remote
- Network
- Batch
- Detached
- Subprocess

Logins are either *interactive* or *noninteractive*. Interactive logins are performed in a series of steps where the system prompts for information and the user provides it. Noninteractive logins are performed by the system and require no user interaction. Table 3–1 lists interactive and noninteractive types of logins.

Table 3–1 Classes and Types of Logins

Login	Type
LOCAL	Interactive
DIALUP	Interactive
REMOTE	Interactive
NETWORK	Noninteractive
BATCH	Noninteractive
DETACHED	Dependent on Parent Process
SUBPROCESS	Noninteractive

The term *interactive,* as used here, differs from an interactive mode process defined by the DCL lexical function F$MODE( ). An *interactive login* consists of user input to the operating system that provides the user name and password (see Section 3.1.2). An *interactive mode process* is one that is declared to be in communication with a person using SYS$COMMAND and may or may not be created by an interactive login. However, all interactive logins result in an interactive mode process.

3.1.1.1 Local Logins

A *local login* is performed by a user from a terminal connected directly to the central processor or to a terminal server that communicates directly with the central processor. Example 3–1 in Section 3.1.2 illustrates a local login.

Local logins are always interactive.

3.1.1.2 Dialup Logins

When you log in to a terminal that uses a modem and telephone line to make a connection to the computer system, you are completing a *dialup login.* You may execute a few additional steps initially, depending on the nature of the terminal concentrator that your system uses. (Since the actual procedure is site dependent, it is not described in this guide. Your security manager can provide you with the necessary details.) However, once you receive the *Username* prompt, the basic elements of the login are the same as those depicted in Example 3–1 in Section 3.1.2.

A dialup login is always an interactive login.

3.1.1.3 Remote Logins

When you log in to a node over the network, you request that node by entering the DCL command SET HOST. This login is known as a *remote login.* The node you reach immediately asks you for a user name and password, like the local login illustrated in Example 3–1 in Section 3.1.2. Informational messages are optional and remain the same.

A remote login is always an interactive login.

3.1.1.4 Network Logins

Network logins are performed for you when you access files stored in a directory on another node or when you initiate some other type of network task on a remote node. Both your current system and the remote system must be nodes in the same network. You use one of the DCL commands that apply over networks, such as COPY (to copy files between nodes in the network) or DIRECTORY (to view files on a remote node). In the file specification, you specify the desired node and an optional access control string, where the access control string includes your user name and password for the remote node. For example, user CRAND has an account on remote node PARIS and enters the following command to get a directory listing of all the files in the [PUBLIC] directory on disk WORK2:

```
$ DIRECTORY PARIS"CRAND password"::WORK2:[PUBLIC]*.*;*
```

Proxy logins represent a special type of network login. With a proxy login, you also gain network access, but you do not have to include an access control string to provide the user name and password for your network login request. Thus, proxy logins have several security implications. First, passwords are never echoed on the terminal where the request originates, which is very desirable. Second, passwords are not passed between systems where they might be intercepted in unencrypted form. Third, this technique removes the temptation to store passwords in command files that would perform the remote access steps.

Proxy logins are described in Section 3.2.2.

All network logins are noninteractive.

3.1.1.5 Batch Logins

A *batch login* is performed for you when a batch process you initiate actually runs. For example, you might submit a job to run after 7:00 p.m. with the following DCL command:

```
$ SUBMIT/AFTER=19:00 PAYROLL.COM
```

When the system prepares to execute PAYROLL.COM, the batch job controller first logs in to the user's account to gain access to the program. This login is classified as a batch login and is noninteractive.

The batch job controller does not need a password to perform the login.

3.1.1.6 Detached Process Login

A *detached process login* occurs as the result of the execution of either the process form of the DCL command RUN or the system service $CREPRC, where the image specified is SYS$SYSTEM:LOGINOUT.EXE, and the options on the service call or the RUN command specify a detached process.

The creator of a detached process login can specify that its type of login is either interactive or noninteractive.

3.1.1.7 Subprocess Login

A *subprocess login* occurs as the result of the execution of either the process form of the DCL command RUN or the system service $CREPRC, where the image specified is SYS$SYSTEM:LOGINOUT.EXE, and the options on the service call or the RUN command specify a subprocess. In addition, a subprocess login results when the DCL command SPAWN executes.

A subprocess login is always a noninteractive login. It always runs under the account of the creator.

3.1.2 Interactive Login Informational Messages

Example 3–1 represents a local login from a terminal directly connected to the system and includes examples of most informational messages.

Example 3–1 Local Login Messages

```
  WILLOW - a member of the Arboretum VAXCLUSTER              ❶ announcement msg

Username:  RWOODS
Password:
    You have the following disconnected process:           ❷ disconnected job
Terminal    Process name    Image name                            messages
VTA52:      RWOODS          (none)
Connect to above listed process [YES]: NO
        Welcome to VAX/VMS Version 5.0 on node WILLOW       ❸ welcome msg
    Last interactive login on Wednesday,  1-AUG-1988 10:20  ❹ last login msg 1
    Last non-interactive login on Monday, 30-JUL-1988 17:39 ❺ last login msg 2
        2 failures since last successful login              ❻ last login msg 3

           You have 1 new mail message.                     ❼ new mail msg

$
```

An announcement message is shown at callout ❶, disconnected job messages at callout ❷, welcome message at callout ❸, and an optional group of last login messages at callouts ❹, ❺, and ❻. If new mail has been delivered, users receive a new mail message similar to the one at callout ❼.

3.1.2.1 Announcement Message

The *announcement message* typically identifies the node (and, if relevant, the cluster) that you have succeeded in accessing. The announcement message is the first visual indication that you have initiated the login process. The announcement message immediately precedes the *Username* prompt. Both the appearance and content of this message can be controlled by the system or security manager redefining the logical name SYS$ANNOUNCE in the site-specific startup command procedure (SYS$MANAGER:SYSTARTUP.COM).

3.1.2.2 Disconnected Job Messages

When you succeed in logging in, you may see messages similar to those in Example 3–1 at callout ❷ reporting *disconnected jobs*. Such messages indicate that a previous login was interrupted prematurely but is available for reconnection. These messages appear only when two conditions exist. First, the terminal where the interruption occurred must have been set up as a virtual terminal to prohibit a process from being disconnected when the line sensed a hangup. Second, during a recent session, your connection to the CPU through that terminal must have been broken. Refer to the *Guide to Maintaining a VMS System* for information on setting up and reconnecting to virtual terminals.

The *disconnected job messages* inform you that your process was disconnected at some time after your last successful login. You are given the option of reconnecting to the old process. Reconnecting returns your process to its state before you were disconnected. If you take the default action or respond to the question with a "YES" answer, you are logged out of your current process as if automatic execution of the DCL command CONNECT/CONTINUE had

been performed for you. If you specify "no" in response to the reconnection question, or you delay too long in responding so that a response period timeout occurs, you remain logged in to your new process, and you lose the ability to connect to the old process. If you have multiple disconnected sessions, you are prompted for the name of the virtual terminal to which you want to reconnect. If you do not want to connect to any of the displayed sessions, enter "NO."

3.1.2.3 Welcome Message

Once you have logged in, you may find a *welcome message* that indicates the software version of VMS that is running and possibly the node name. If the system manager chooses, an entirely different message may appear. The system manager can also choose to suppress the message entirely.

3.1.2.4 Last Login Messages

Immediately following the welcome message are three messages providing information about the last successful login. These *last login messages* are optional. They are enabled or disabled as a group.

If the messages are enabled, you receive from one to three of the following messages:

- Last successful interactive login message—provides the time of the last completed login for a local, dialup, or remote login. (Note that logins from a subprocess whose parent was one of these types are not included in the count.) An example appears at callout ❹ in Example 3–1.
- Last successful noninteractive login message—provides the time the last noninteractive login successfully completed. Noninteractive logins refer to batch or network process logins. An example appears at callout ❺ in Example 3–1.
- Number of login failures—if any attempts have been made to log in and have failed because of an incorrect password, they are recorded in a count displayed in this message at the next successful login. To call your attention, a bell rings after the message appears. (An incorrect password is the only source of login failure that is counted.) An example appears at callout ❻ in Example 3–1.

All three types of login message can be displayed at the next successful login, whereupon the values for the last successful login and the number of login failures are reset. If you always access your account interactively and never specify an incorrect password in your login attempts, you might never see the last successful noninteractive login and login failure messages.

3.1.2.5 Message Suppression

A security manager can suppress the announcement and welcome messages, which include node names and operating system identification. Because login procedures differ according to operating system, it is more difficult to log in without this information.

Sites with medium- or high-level security needs can display the last login success and failure messages. These messages may indicate break-in attempts and should be checked regularly. They may also be a deterrent to potential illegal users by indicating that the system is monitoring logins.

The disconnected job messages appear less frequently and only under special circumstances. Virtual terminals must be enabled on your system, and the terminal whose connection was broken prior to a logout during your last session must also have been set up as disconnectable. The security manager can disable this function by changing the setup on terminals and disabling virtual terminals on the system. The ability to reconnect is generally desirable and offers no special problems for system security.

3.1.3 Introduction to Passwords

The VMS operating system requests a password when you log in. Passwords are strings of characters that users must specify when they log in to show authorization to use the account. To preserve the secrecy of passwords, terminals do not echo password characters as they are entered. Proper administration of passwords is critical to the security of a system.

There are several types of passwords on the VMS system. Most users need to provide a *user password* when they log in. Some users also need to provide a *system password* to gain access to a particular terminal before logging in with their user password. Users on systems with high security requirements need to provide *primary passwords* and *secondary passwords*. These passwords are described in the following sections.

The VMS operating system applies a *one-way encryption algorithm* to all passwords as it stores them. Encryption refers to a method of encoding information in an effort to conceal it. One-way encryption algorithms do not use a key. Thus, even if a malicious user succeeds in obtaining the encryption algorithm and the encoded password, that user could deduce the actual password only by trying all possible input values.

When a user specifies a password, VMS always runs the submitted password through the encryption algorithm before attempting to match the derived value with the stored value. If the two encoded values match, VMS grants access to the user.

3.1.3.1 System Passwords

System passwords control access to particular terminals and are required at the discretion of the security manager. They are usually necessary to control access to terminals that might be targets for unauthorized use, such as dialup and public terminal lines.

Your security manager will tell you if you must specify a system password to log in to one or more of the terminals designated for your use. Your security manager will also provide you with the current system password, how often it changes, and how to obtain the new system password when it does change.

Specify a system password by pressing the RETURN key until the terminal sets the baud rate (called *autobauding*) and responds with the recognition character, which is normally a bell. If your terminal has been set with the /NOAUTOBAUD characteristic, you will only press the RETURN key once. At that point you type the system password followed by RETURN. There will be no prompt and no echo of the characters you type. When you succeed in entering the correct system password, you will receive the system announcement message, if there is one, followed by the *username* prompt. If you fail to specify the correct system password, you can try repeatedly. There is no notification given that you have entered an incorrect password. Thus, you might initially think the system is malfunctioning unless you know that a system password is required at that terminal.

3.1.3.2 User Passwords

Following are the four types of user accounts available on VMS systems:

- *Open accounts* require no password; the password is null. Users of open accounts are the only users who do not need to enter user passwords. The user is not prompted for a password and can begin entering commands immediately.
- *Captive accounts* may require a password. A captive account limits user operations.
- Accounts that always require passwords and prohibit the user from changing the password. The password is locked by setting the LOCKPWD flag in the User Authorization File (UAF). By locking the password, the security manager controls all changes made to the password.
- Accounts secured with passwords that are changed periodically by the user or security manager. Because this account type is the most commonly used, it is the type referred to in this guide.

Typically, when you learn an account has been created for you on the system, you are told whether or not a user password is required. If user passwords are in effect, you will be told to use a specific password for your first login. This password has been placed in your record in the UAF with other pertinent information about how your account can be used.

Often, your first name is used as your first password. This practice is so well known that it is undesirable from a security standpoint. Although you are requested to change your password as one of your first actions after logging in, your account remains highly vulnerable until you do so. If there is a time lapse from the time your account is created until your first login, other users might log in to your account successfully, gaining a chance to damage the system. Similarly, if you neglect to change the password or are unable to do so, the system remains vulnerable. Possible damage depends largely on what other security measures are in effect.

It is inadvisable to have accounts on the system where the password can be easily guessed. Avoid using the following as passwords:

- Your name
- The name of a family member or loved one
- The name of a pet
- The make of your favorite type of automobile
- The name of your hometown
- The name or make of your boat
- Any name associated with your work, such as your company, special project, or group
- Any other item that bears a strong personal association to you

At the time your account is opened, you should also be told a minimum length for your passwords and whether you will be able to choose your new password or must let the system generate the password for you.

3.1.3.3 Changing Your Password

Change your password using the DCL command SET PASSWORD. This command supports two modes: changes entered by the user and automatic password generation by the system. The system manager can require that you use the automatic password generator when changing your password or can allow you to select the method of changing your password.

User-Selected Passwords

To change your password, invoke the SET PASSWORD command with no qualifiers. The command prompts you to provide the old password and then requests that you enter the new password. As a final step, the system asks you to enter the new password one more time for verification. If you fail to enter the old password correctly or do not enter the same password twice as the new password, the password is not changed. If you succeed in these three steps, there is no notification. The command terminates, and your password is changed. (Section 3.1.3.5 describes in more detail the consequences of entering a new password choice that does not meet the minimum password length requirement.)

You will have learned the minimum number of characters required for your VMS passwords. This characteristic is part of your user authorization record. When you designate a new password, the system enforces your minimum password length requirement. You can enter a password choice that is equal to or longer than the minimum, but any shorter password choices will be rejected.

Automatically Generated Passwords

If your system security manager has decided that you must let the system generate the password for you automatically, use the DCL command SET PASSWORD/GENERATE to change your password. If you attempt to use the SET PASSWORD command without the /GENERATE qualifier, the system inserts it.

Automatic password generation produces a list of password choices made up of random sequences of characters. The sequence resembles English words to make it easy to remember but is unusual enough to be difficult for outsiders to guess. Because system-generated passwords vary in length, they become even more difficult to guess.

The following example illustrates a user requesting automatic password generation. The minimum password length for this user has been set to 6 in the UAF.

```
$ SET PASSWORD/GENERATE=8
Old password:

apsjawpha     aps-jaw-pha
oorsoult      oor-soult
guamixexab    gu-a-mix-ex-ab
impsapoc      imps-a-poc
ukchafgoy     uk-chaf-goy

Choose a password from this list or press RETURN to get a new list
New password:

Verification:
$
```

In the preceding example, the user requests the automatic password generator to provide password choices with a minimum length of eight. The user correctly specifies the old password and presses RETURN. The system responds with a list of five password choices ranging in length from eight to ten characters.

To the right of each password choice is a representation of the same word divided into its syllables. Usually the password that is easiest to pronounce is easiest to remember and, therefore, the best choice.

Next, the system reminds the user that it is possible to request a new list by pressing the RETURN key in response to the prompt for a new password. However, in this case, the user enters one of the first five possible passwords, followed by RETURN. The system recognizes that this password is one provided by the automatic password generator and responds with the *Verification* prompt. The user enters the new password again followed by RETURN. The system changes the password and responds with the DCL prompt.

On systems where you are not required to use the password generator, you are strongly encouraged to use it on your own to promote the security of your system. A disadvantage of automatic password generation is the possibility that users may not remember their password choices. However, if you dislike all the password choices in your list or think none will be easy to remember, you can always request another list.

A more serious drawback is the potential disclosure of password choices from the display the command produces. To protect your account, perform automatic password changes in private. If you perform the change on a video terminal, erase the display of the password choices from the screen after the command completes. If you use a printing terminal, properly dispose of all hardcopy output. If you later realize that you failed to protect your password in these ways, change your password immediately. Depending on site policy or your own judgment concerning the length of time your account was exposed, you might decide to notify your security manager that a security breach could have occurred through your account.

Note: The password generator uses basic syllabic rules to generate words, but has no real knowledge of any language. As a result, it may unintentionally produce words that are offensive.

There is no restriction on how many times you can change your password in a given period of time. There is a maximum period of time that you can retain the same password. This maximum period is dictated by the password lifetime characteristic in your UAF record set by the system manager.

3.1.3.4 Password Expiration Time

Changing passwords on a regular basis promotes system security. The VMS operating system includes automatic password expiration as one of its security features.

As you approach the expiration time of your password, you receive an advance warning message. The message first appears each time you log in five days prior to the expiration date. The message appears immediately below the new mail message and sounds the bell character on your terminal to attract your attention. The message indicates that your password is expiring, as follows:

```
WARNING -- Your password expires on Thursday 19-JUL-1988 15:00
```

If you fail to respond to this warning, you will receive the following message when you log in after your password expires:

```
Your password has expired; you must set a new password to log in

Old password:
```

The system then prompts you for a new password, or, if automatic password generation is enabled, asks you to select a new password from those listed. You can abort the login by pressing CTRL/Y. You will be prompted to change your password on your next login attempt.

If secondary passwords are in effect for your account (see Section 3.1.3.7) and both primary and secondary passwords are expired, you are prompted to change both passwords. If you change the primary password and press CTRL/Y before changing the secondary password, the login aborts and no password change is recorded.

If the system manager decides not to force you to change your expired passwords upon logging in, you receive one final warning when you log in after your password expires, as follows:

```
WARNING -- Your password has expired; update immediately with SET PASSWORD!
```

At this point, if you do not change the password, or if the system fails before you have the opportunity to do so, you must see your system manager to regain access.

Note that you cannot specify your old password as your new password.

3.1.3.5 Minimum Password Lengths

Your security manager can establish a minimum length for your passwords by specifying a value in your UAF record. Use of longer passwords makes penetration more difficult. The VMS system encourages minimum password lengths in the range of 6 to 10 characters. You can choose a password as long as 31 characters, but entering long passwords is too cumbersome and error-prone to be practical.

If you password choice is too short, you receive the following message:

```
%SET-E-INVPWDLEN, invalid password length - password not changed
```

3.1.3.6 Selecting Secure Passwords

As stated, adequate length makes passwords more secure. Avoid selecting passwords from a dictionary or from your native language.

The most secure passwords include both digits and letters. For example, if you could choose a six-character password using letters only, you would find there are 300 million combinations. However, when you allow that same six-character password to include digits, you increase the number of combinations to 2 billion. Consider including digits in your password selection.

3.1.3.7 Primary and Secondary Passwords

VMS can provide an additional level of security on user accounts by requiring the use of secondary passwords in addition to primary passwords. Your security manager decides whether to adopt this practice for your account at the time the account is created. When primary and secondary passwords are required, you must log in by first specifying both passwords correctly.

In some cases one user knows both passwords and uses them to log in. However, the more typical case involves two users, where the primary user does not know the secondary password. This arrangement is designed to facilitate controlled logins. By requiring the presence of a supervisor or other key person at login time, there is added security.

Dual passwords can also help control the actions taken after a login. For certain applications, it may be desirable for another person to remain present while the account is in use.

A login requiring primary and secondary passwords might appear as follows:

```
  WILLOW - a member of the Arboretum VAXCLUSTER

Username: RWOODS
Password:
Password:
        Welcome to VAX/VMS Version 5.0 on node WILLOW
$
```

As with a single password login, there is a limited amount of time allotted for the entire login. If the entry of the secondary password is not completed in time, the login will time out.

Requiring a secondary password is time-consuming and inconvenient. It is justified only at sites with maximum security requirements. An example of an account that justifies dual passwords would be one that bypasses normal access controls to permit emergency repair to a database.

Primary and secondary passwords can be changed independently, but both are subject to the same change frequency since they share the same password lifetime. Minimum password length applies to both passwords.

To change the primary password, follow the steps in Section 3.1.3.3. To change the secondary password, use the SET PASSWORD/SECONDARY command. You are prompted to specify the old secondary password and the new secondary password, just as in the procedure for changing the primary password.

3.1.3.8 Avoiding Programs That Steal Passwords

Beware of attempting a login from a terminal that is already turned on. You might be revealing your password to a program specially designed to steal passwords. This precaution is particularly relevant when you are working in a public terminal room.

A *password grabber* program is a special program that displays an empty video screen, a screen that appears to show the system has just been initialized after a crash, or a screen that shows a nonexistent logout. When you attempt to log in, the program runs through the normal login sequence so you think you are entering your user name and password in a normal manner. However, once the program receives this key information and passes it on to the perpetrator, it displays a login failure. You may think you mistyped your password and be unaware that you have just revealed this vital information.

Your security manager will instruct you on how to eliminate this possibility. You may be advised to press the BREAK key before logging in. Pressing the BREAK key invokes the *secure terminal server* feature for the terminal, if it has been enabled by the security manager. The secure server ensures that the VMS login program is the only program able to receive your login.

Do not leave your terminal unattended after you log in. You may think the system failed and came back up again, when actually someone has loaded the password stealing program. Even a terminal that displays an apparently valid LOGOUT message may not reflect a normally logged out process.

Check your last login messages routinely. The password stealing program cannot actually increase the login failure count, although it looks like a login failure to you. Be alert for login failure counts that do not appear following your failure, or that are one less than the number you experienced. If you observe this or any other anomalous failure during a login, change your password immediately and notify your security manager.

3.1.3.9 Protecting Your Password

Illegal system accesses involving the use of a correct password are more often traced to disclosure of the password by its owner than to surreptitious discovery. It is vital that you do not reveal your password to anyone. Do not give it to friends, store it in a file, or send it in a mail message.

Sometimes inadequately protected files include character strings that are likely to appear in conjunction with actual passwords. Browsers might search your files for the three-character string specific to network access control strings—a quotation mark followed by two colons ("::). This string immediately follows the username and password specification for network file accesses. A browser may also search your files for the string "password". For example, a careless user may reveal the actual password in a sentence like the following:

```
My password is GOBBLEDYGOOK.
```

Do not use the same password for accounts on different systems. An unauthorized user will try one password on every system where the same user has an account. The account that first reveals the password may hold little of interest, but another account might yield more information or more privileges, ultimately leading to a far greater security breach.

If you hold accounts on multiple systems, there are several special considerations for your password, as follows:

- If any of the systems requires high security, use a unique password on that system.
- If any of the systems uses non-DIGITAL computers, use unique passwords on your accounts.
- If the systems are either all VMS systems or employ password encryption, you can use the same password for each system, as long as you observe all other guidelines regarding good password choices.
- If a system does not use password encryption, select a unique password for that system.

3.1.3.10 Summary of Password Guidelines

To summarize, you can best protect your password by observing the following guidelines:

- Select reasonably long passwords that cannot be easily guessed. Avoid using words in your national language that would appear in a dictionary. Consider including digits in your passwords. Alternatively, let the system generate passwords for you automatically.
- Never write down your password.
- Give your password to other users only under special circumstances. Change it immediately after the need for sharing has passed.
- Do not include your password in any file, including the body of an electronic mail message. (If anyone else reveals a password to you, delete the information promptly.)
- Before you log in to a previously turned on terminal, invoke the secure terminal server feature (if enabled) with the BREAK key.
- Unless you share your password, change it every three to six months. DIGITAL warns against sharing passwords. If you share your password, change it every month.
- Change your password immediately if you have any reason to suspect it might have been discovered. Report such incidents to your security manager.
- Do not use the same password for your accounts on multiple systems.

3.1.4 Account Expiration Times

When your account is created, the security manager may decide to specify a period of time after which the account will lapse (for example, if you will only need the account for a specific purpose for a limited time). At universities, student accounts are typically authorized for a single semester at a time. Expired accounts automatically deny logins.

Users receive no advance warning message prior to the expiration date, so it is important to know in advance what your account duration will be. The account expiration resides in the UAF record, which can be accessed and displayed only through the use of the VMS Authorize Utility by users with the SYSPRV privilege or equivalent—normally your system or security manager.

When your account expires, you receive an authorization failure message at your next attempted login. If you need an extension, follow the procedures defined at your site.

3.1.5 Causes of Login Failures

Possible causes for login failure are listed in Table 3–2.

Table 3–2 Causes of Login Failure

Login Failure	Cause
No response from the terminal	Attempting to use a defective terminal. The device requires a system password.
No response from any terminal	Attempting to log in when the system is down.
No response from terminal to your entry of system password	The system password changed and you were not notified.
Message: User authorization failure	Mistyping the user name or password. Attempting to use an expired account. Attempting to use an expired password.
Message: Not authorized to log in from this source	The attempted class of login (LOCAL, DIALUP, REMOTE, INTERACTIVE, BATCH, or NETWORK) is prohibited.
Message: Not authorized to log in at this time	The day of the week or hours of the day are not permitted for you for this class of login.
Message: User authorization failure (and no known user failure occurred)	An apparent break-in has been attempted at the terminal using your user name, and the system has temporarily disabled all logins at that terminal by your user name.

The following sections describe login failures.

3.1.5.1 System Password Failures

You cannot log in if the terminal you attempt to use requires a system password and you are unaware of the requirement. As Section 3.1.3.1 explains, some systems require that a system password be entered from a particular terminal before anyone can log in at the terminal. (This is an option the security manager may choose to implement.) There is no warning message or response, so the system appears to be down. All attempts at logging in will fail until the system password is entered. If you suspect that this is the problem, try logging in at another terminal.

If you have been directed to use a terminal that requires a system password and know the password, perform the steps described in Section 3.1.3.1. If your attempts fail, it is possible that the system password has been changed. Move to a different terminal that does not require a system password or request the new system password.

3.1.5.2 Login Class Restrictions

If you attempt a class of login that is prohibited in your UAF record, your login will fail. For example, your security manager may have restricted you from logging in over the network. If you attempt a network login, you receive a message telling you that you are not authorized to log in from this source.

Your security manager can restrict your logins to include or exclude any of the following classes (discussed in Section 3.1.1): LOCAL, REMOTE, DIALUP, BATCH, or NETWORK. The general name INTERACTIVE is useful to include or exclude all three of the classes LOCAL, REMOTE, and DIALUP using one expression.

3.1.5.3 Shift Restrictions

Another cause of login difficulty is failure to observe your shift restrictions. The security manager can restrict your logins to certain times of the day and certain days of the week. These restrictions are imposed on classes of logins. The security manager may apply the same work-time restrictions to all classes of logins or choose to place different restrictions on different login classes. If you attempt a login during a time prohibited for that login class, your login fails, and you are notified that you are not authorized to log in at this time.

When shift restrictions apply to batch jobs, jobs you submit that are scheduled to run outside your permitted work times will not be run. The system does not automatically resubmit such jobs during your next available permitted work time. Similarly, if you have initiated any kind of job and attempt to run it beyond your permitted time periods, the job controller will abort the uncompleted job when the end of your allocated work shift is reached. This job termination behavior applies to all jobs.

3.1.5.4 Dialup Login Failures

Your security manager can control the number of opportunites you are given to enter a correct password during a dialup login before the connection is automatically broken.

If your login fails and you have attempts remaining, press RETURN and try again. You may do this until you succeed or reach the limit. If the connection is lost, you can redial the access line and start again.

The typical reason for limiting the number of dialup login failures is to discourage unauthorized users attempting to learn passwords by trial and error. They already have the advantage of anonymity because of the dialup line. However, limiting the number of tries for each dialup does not necessarily stop this kind of break-in attempt. It only requires the would-be perpetrator to redial and start another login.

3.1.5.5 Break-In Evasion Has Been Activated

If anyone has made a number of failed attempts to log in at the same terminal with your user name, the system may respond as though a break-in attempt is in progress. That is, the system concludes that someone is attempting to gain illegal access to the system using your user name. As a result, the system has disabled even your valid logins on that terminal for a certain period of time to frustrate the would-be perpetrator.

At the discretion of your security manager, *break-in evasion* measures may be in effect for all users of the system. The security manager controls how many password attempts are allowed over what period of time. Once break-in evasion tactics are triggered, you will be unable to log in to the terminal—even with your correct password—during a defined interval. Your security manager can tell you how long you must wait before reattempting the login, or you can move to another terminal to attempt a login.

If you suspect that break-in evasion is preventing your login, and you have not personally experienced any login failures, you should immediately reach your security manager. Together you should attempt another login, checking the message that reveals the number of login failures since the last login, to confirm or deny your suspicion of break-in attempts. (If your system does not normally display the login message, your security manager can use the VMS Authorize Utility to examine the data in your UAF.) With prompt action, your security manager may locate someone attempting logins at another terminal.

3.2 Network Security Considerations for Users

This section describes several ways to help make network access more secure. Topics described are access control strings in file specifications and command procedures, proxy logins, and proper use of the VMS Mail Utility.

3.2.1 Network Access Control Strings

Network access control strings are designed to be included in the file specifications of DCL commands working across the DECnet–VAX network. They permit a user on a local node to request an operation using a file on a remote node. An access control string consists of the user name for the remote account and the user's password enclosed in parentheses, as shown in the following example:

```
NODE"username password"::disk:[directory]file.typ
```

Because access control strings include sufficient information to allow someone to break in to the remote account, they create serious security exposure.

To protect access control string information, avoid revealing the information on either hardcopy or video terminals. Do not place networking commands in command procedures where they would be likely targets for discovery. The syntax that requires the user name and password to be placed within quotation marks and followed by two colons makes searches for passwords in insufficiently protected files easy. A password usually precedes the three-character access control string terminator ("::). If you must put networking commands that include access control strings in your command procedures, provide these files with optimum file protection using the techniques described in Chapter 4.

You might prefer to use proxy login accounts to avoid the need for access control strings. Section 3.2.2 explains proxy logins.

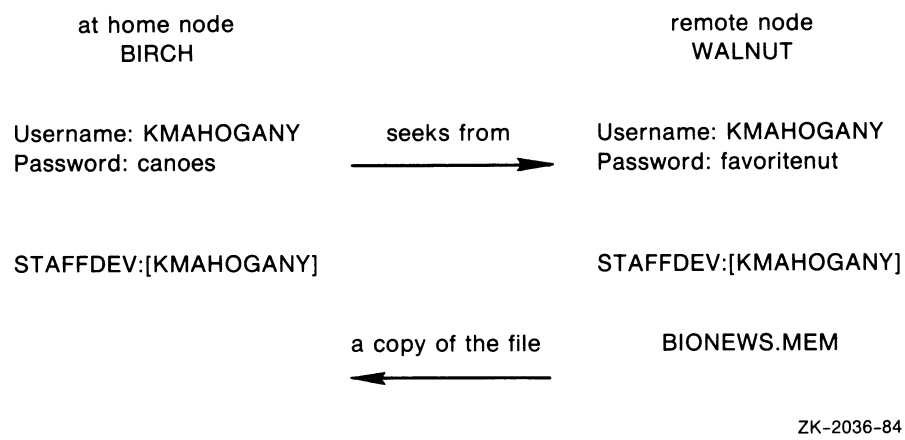
3.2.2 Proxy Logins

Proxy logins allow users to access files across a network without specifying user name or password in an access control string.

Before a user can enter a request initiating a proxy login, the system or security manager at the remote node must create a proxy account for that user. Proxy accounts, like regular accounts, are created with the VMS Authorize Utility. However, proxy accounts also use the *network proxy authorization file*, NETPROXY.DAT, to identify which remote users are allowed access to proxy accounts on the system.

The following examples illustrate the differences between a normal network login request and a proxy login request. In the first case, the user has two user accounts, one on the node BIRCH with the password "XYZ123ABC" and one on the node WALNUT with the password "A25D3255". The user has logged into BIRCH and wants to copy the file BIONEWS.MEM over from the default device and directory of the account on the node WALNUT. The situation is diagrammed in Figure 3–1.

Figure 3–1 Illustration of File Sharing over a Network



The user enters the following command:

```
$ COPY WALNUT"KMAHOGANY A25D3255"::BIONEWS.MEM  BIONEWS.MEM
```

Notice that since the password of the KMAHOGANY account on WALNUT echoes, it is revealed to anyone who observes the screen.

Note: If you use passwords to access files across a network, remember to clear the screen and empty the recall buffer with the DCL command RECALL/ERASE when the network job is completed. This prevents others from viewing the previously entered password using the DCL commands CTRL/B or RECALL/ALL.

In the next example, the security manager at the node WALNUT also maintains an account for Kay Mahogany with the username KMAHOGANY and the default device and directory of STAFFDEV:[KMAHOGANY]. However, this time the security manager at WALNUT authorizes user BIRCH::KMAHOGANY to perform proxy logins into the WALNUT::KMAHOGANY account. As the owner of the account on WALNUT, Kay has a password, but she will not need to specify it in her access control string when she performs network commands from BIRCH, because the system will perform a proxy login into her account. To copy the file BIONEWS.MEM from the default device and directory of the KMAHOGANY account on WALNUT, Kay Mahogany enters the following COPY command:

```
$ COPY WALNUT::BIONEWS.MEM   BIONEWS.MEM
```

Since no access control string was provided in the network copy command, the VMS operating system attempts to complete the operation using proxy logins. If proxy access has been set up for remote user BIRCH::KMAHOGONY, a network proxy login is performed on node WALNUT granting BIRCH::KMAHOGANY proxy access to the KMAHOGONY account. There is no exchange of passwords.

This is an example of a single user proxy login account; Kay is the only remote user allowed to access the account. However, it is also common practice to authorize groups of users from foreign nodes to share in the use of single home node accounts, as in the following example.

The security manager at the node WALNUT creates a general access account with the user name GENACCESS that permits only network logins. The default device and directory are STAFFDEV:[BIOSTAFF]. Next, the security manager authorizes a number of remote users for proxy logins to this account and includes Kay Mahogany on node BIRCH as one of those authorized users. The owner of the general access account on WALNUT has a password, but none of the remote users like Kay Mahogany need to know it. (Again, this is deliberate, to protect the account.) To copy the file BIONEWS.MEM from her own directory on the default device accessed through the GENACCESS account on WALNUT, Kay Mahogany enters the following COPY command:

```
$ COPY WALNUT::[KMAHOGANY]BIONEWS.MEM   BIONEWS.MEM
```

Notice that this command specifies an explicit directory for node WALNUT. This is only necessary because the file BIONEWS.MEM resides in STAFFDEV:[KMAHOGANY] while STAFFDEV:[BIOSTAFF] is the default device and directory for the proxy login account GENACCESS on node WALNUT. Note that the protection for the file BIONEWS.MEM must permit access to the GENACCESS account, or the example fails.

Observe how this arrangement also succeeds in eliminating the need to forward the password as a string. This is the key characteristic of proxy logins, whether the accounts are set up for single remote users or groups of remote users. Had the file BIONEWS.MEM been moved to the directory [BIOSTAFF], the two copy commands illustrating proxy logins could have been identical.

Thus, while proxy logins require more setup effort on the part of system managers, they provide more secure network access and involve much simpler procedures for the users.

3.2.2.1 Multiple Proxy Accounts

Security managers can allow remote users access to one default proxy account and up to 15 other proxy accounts. If a remote user has access to more than one proxy account and wants to copy files over the network using a proxy account other than the default, the name of the proxy account must be specified in the access control string. For example, suppose remote user Kay Mahogany in the previous section has proxy access to the GENACCESS, PROXY2, and PROXY3 accounts and the default proxy account is GENACCESS. To copy the file BIONEWS.MEM from the default device and directory on node WALNUT using the PROXY2 proxy account, Kay Mahogony enters the following COPY command:

```
$ COPY WALNUT"PROXY2"::BIONEWS.MEM BIONEWS.MEM
```

3.2.3 Using the VMS Mail Utility

The VMS Mail Utility has default file protection to discourage mail tampering. However, the Mail Utility is not completely tamperproof. Anyone with sufficient privilege can change protection and access mail files.

Therefore, use discretion both in the content of your mail messages and in the selection of your audience. Never reveal your password or send details about how to use your account. You have no control over information in a mail message once you have sent it.

3.3 Logging Out of the System

When you leave your terminal on line and your office open, you have effectively given away your password, your privileges, and left your files and those of the other members of your group unprotected. Any user can easily and quickly transfer all files accessible through your account. A malicious insider could rename and delete your files and any to which you have WRITE access. If you have special privileges, especially SETPRV, a malicious user can do major damage.

Therefore, establish criteria for when to log out of your system. Log out when you leave your office even for a brief period of time. Leaving a terminal on line represents one of the greatest sources of inside break-ins.

3.3.1 Logging Out from Video Terminals

There are several steps you may want to consider each time you log out. Information left on your terminal screen at logout can vary. At sites with medium-level security concerns, it may be important to leave nothing but the logout message on your screen, as follows:

- If your terminal is a VT100 series terminal, clear the screen by using the setup feature of your terminal. (Press the SET-UP key, then the key marked for reset (the 0 key), followed by the RETURN key.)
- If your terminal is in the VT200 or VT300 series of terminals, you can accomplish this by pressing the SET-UP key, then selecting the item from the resulting menu that corresponds to CLEAR DISPLAY.

Once the screen has cleared, and the DCL prompt returns, you can enter the DCL command LOGOUT. Since your screen is clear, the cursor is positioned at the top of the screen. The only information remaining is your logout command and the logout completion message, as follows:

```
$ LOGOUT
  RDOGWOOD     logged out at 14-AUG-1988 19:39:01.43
```

At high-security sites, it is common practice to turn off your video terminal every time you log out because the logout message reveals a currently active user name. When users log off after a remote login, the name of the node they return to after the remote logout is also revealed. When a user has accessed multiple accounts remotely over the network, the final sequence of logout commands reveals all the nodes and the user names that are accessible to the user on each node, with the exception of the name of the furthest node reached. To those who can recognize the operating system from the prompt or a logout message, this will also reveal the operating system.

If the system fails before you log out, there may be important information left on your screen. To avoid this, turn your video terminal off and then on. Then wait for the system initialization message or your first chance to reaccess the terminal concentrator. If you plan to leave the area, do not turn the terminal on again until you return. Observe the precautions described in Section 3.1.3.8 to avoid password grabbers.

3.3.2 Logging Out from Hardcopy Terminals

When you need to log out from a hardcopy terminal, enter the DCL command LOGOUT. If you have performed remote logins, you must log out of each node. Properly remove, file, or dispose of all hardcopy output that might reveal security information. Your security manager should provide direction on preferred procedures. Many sites use paper shredders or locked receptacles for this purpose. Handle output that you plan to save just as carefully.

If the system fails before you log out, dispose of the hardcopy output. Turn your terminal off if you will not be present when the system is initialized.

3.3.3 Logging Out from Disconnected Processes

To conserve system resources, perform the following steps:

- Enter the DCL command SHOW USERS to determine if you have other disconnected jobs.
- Enter the DCL command CONNECT/CONTINUE to log out of the current process. Connect back through each of the associated virtual terminals (as noted by the terminal prefix of VTA) until you have reached the last existing process.
- Enter the DCL command LOGOUT.

If you do not perform these steps, the system automatically removes your disconnected processes after a certain interval. Performing these steps saves system time.

3.3.4 Logging Out from a Dialup Login

Your security manager may request that you always use the /HANGUP qualifier with your final LOGOUT command when you log out from a dialup login and anticipate no further immediate use of the line. The use of the /HANGUP qualifier on a LOGOUT command directs VMS to automatically break the connection to the dialup line after the logout. No one can take advantage of an open access line; it is necessary to know the access number and personally redial. This is especially important if the dialup line you use is in a public area or where someone might use the terminal after you.

The /HANGUP qualifier will work for all terminals that have been set up with the HANGUP terminal characteristic. (The system manager specifies the HANGUP characteristic with the DCL command SET TERMINAL/PERMANENT/HANGUP.)

This practice also saves resources, since dialup lines are limited.

3.4 Summary of Good User Practices

This chapter has presented many suggestions for users to help maintain a secure system. Although many security features are implemented by the security manager as requirements for all users, there are many ways users can contribute to system security. The list following reviews voluntary security actions. Included are cross-references to parts of this chapter where each topic is discussed.

- Protect your password (Section 3.1.3.10).
- Check your last login messages each time you log in, and report any unexplained messages to your security manager (Section 3.1.2.4).
- Lock up and log out when you leave your terminal and area (Section 3.3).
- Use the /HANGUP qualifier on your final LOGOUT from a dialup line (Section 3.3.4).
- Properly dispose of hardcopy output from your terminal (Section 3.3.2).
- Turn off your video terminal to erase revealing displays (Section 3.3.1).
- Use proxy logins where possible (Section 3.1.1.4).

4 File Protection Features

File protection mechanisms are extremely important tools for enhancing system security. This chapter explains file protection mechanisms and their use.

The VMS operating system offers two primary protection mechanisms. The first, *standard UIC-based protection,* is based on the user identification code (UIC) and is applied to all user files. It controls access to files according to the user categories SYSTEM, OWNER, GROUP, and WORLD.

The second file protection mechanism uses *access control lists (ACLs),* which employ a more refined level of protection on files than that available with UIC-based protection. ACLs can be used to grant or deny file access to individual users or groups of users, independent of the UIC.

ACLs are important file protection tools available to all VMS users and are generally used at sites with medium to high security requirements. ACLs are also prevalent in environments with complex patterns of file sharing. As security requirements increase, so does the use of ACLs.

4.1 How the System Determines Access

There is an interaction between ACLs, the UIC-based protection code, and user privileges that determines the outcome every time a user requests access to an object. This section introduces the order in which VMS evaluates each of these components.

The VMS operating system performs the following steps to determine if a user is allowed access to a particular object:

1. If an object has an associated ACL, the system uses that ACL to determine whether the user should gain access to the object. If the ACL grants the requested access to the user, access is given and all further testing stops. If the ACL denies access, the system uses the SYSTEM and OWNER fields of the UIC-based protection to determine if the user is allowed access. If the ACL does not explicitly grant or deny the user access, the system uses UIC-based protection to determine access.
2. If an object does not have an associated ACL, the system uses UIC-based protection to determine access. (See Section 4.2.3.)
3. The GRPPRV, SYSPRV, READALL, and BYPASS privileges amplify the privilege holder's access to objects under certain circumstances. (See Section 4.2.5.)

Following is a user-access summary:

- ACLs are always evaluated first.
- When an ACL fails to specifically grant access, UIC-based protection is checked.

- When an ACL specifically denies access, the user can still acquire access by being a member of the SYSTEM or OWNER categories and thereby eligible for access through them, or by possessing privileges.
- Users with certain system privileges may be entitled to access regardless of the protection offered by the ACLs or the UIC protection code.

4.2 Standard UIC-Based Protection

When security managers create accounts, they perform two actions that affect the UIC-based protection:

- They establish each account with a standard default protection code for all files the user creates in the initial top-level directory.
- They assign each user membership in a group.

Your default protection code and group assignment may be sufficient for all files that you work with. Use the DCL command SET PROTECTION if you need to change the protection on certain files or SET PROTECTION/DEFAULT if you need to change the default protection for all new files that you create. Section 4.2.4 describes the syntax of the SET PROTECTION command.

Each user of the system has a UIC defined in the system user authorization file (UAF). Each system object (such as a file) also has an associated UIC, defined to be the UIC of its owner, and a protection code that defines who is allowed what type of access. The relationship between the UIC of the user and the UIC of the object controls access to the object.

4.2.1 UICs and Protection

UIC-based protection is determined by the UIC of the owner of the object and the protection code defined for the object. (Section 4.2.3 defines how the VMS operating system groups users according to UIC.)

UIC-based protection controls access to objects such as files, directories, and volumes. A volume refers to a mass storage medium mounted on a device. For example, disk packs and reels of magnetic tape are called volumes when mounted on disk and magnetic tape drives. For disk volumes, the system provides protection at the file, directory, and volume levels. For magnetic tape volumes, the system provides protection only at the volume level.

Thus, in addition to protecting the files on mounted disk volumes, the VMS system provides overall volume protection for disks and magnetic tapes. This volume protection is coded into the home block of the disk or magnetic tape. (The home block is the section of the index file that contains access and identification information for the volume.) For more information about setting volume protection characteristics for disks and magnetic tapes, see the *Guide to Maintaining a VMS System* and the descriptions of the DCL commands INITIALIZE, MOUNT, and SET VOLUME in the *VMS DCL Dictionary*.

VMS also provides protection for record-oriented devices such as terminals, line printers, mailboxes, and special-purpose devices. See the description of the SET PROTECTION/DEVICE command in the *VMS DCL Dictionary* for information on how to apply protection to record-oriented devices.

4.2.2 Specifying UICs

The system manager assigns a UIC to each user with the VMS Authorize Utility. A UIC has two formats: numeric and alphanumeric. When a DCL command requires a UIC specification, you can use either format to refer to a user.

For example, the following UIC specifications could all be valid for the user JONES:

```
[360,031]
[JONES]
[GROUP1,JONES]
```

4.2.2.1 Numeric Format UICs

A UIC in numeric format contains a group number and a member number in the following format:

```
[group,member]
```

The brackets are required in the UIC specification. The group number is an octal number in the range of 1 through 37776; the member number is an octal number in the range of 0 through 177776. You can omit leading zeros when you are specifying group and member numbers.

4.2.2.2 Alphanumeric Format UICs

A UIC in alphanumeric format consists of a member name and, optionally, a group name in the following format:

```
[member]
```

```
[group,member]
```

The brackets are required in the UIC specification. The group and member names can each contain up to 31 alphanumeric characters and must contain at least one alphabetic character. The names can include the characters A through Z, dollar signs ($), underscores (_), and the numbers 0 through 9.

4.2.2.3 UIC Translation and Storage

Regardless of the format you use, the system translates a UIC to a 32-bit value that represents a group number and a member number; the high-order 16 bits contain the group number, and the low-order 16 bits contain the member number. When translating an alphanumeric UIC such as [J_JONES], VMS equates the member part of the alphanumeric UIC to both the group and member parts of a numeric UIC. The resulting 32-bit numeric UIC is kept in the *system rights database*, which is a file containing information pertaining to the access rights and attributes associated with identifiers and the holders of those identifiers.

This method of storing alphanumeric UICs dictates that member names must be unique, and that no member can participate in more than one group. That is, each member name must be unique for each user on the system. For example, you could not have the two UICs [GROUP1,JONES] and [GROUP2,JONES] on the same system because the member JONES can have only one associated numeric UIC.

A group name is associated with the group portion of a UIC. When the system translates an alphanumeric UIC that includes both a group and a member name, the system obtains the longword integer associated with the member and checks the group name against the member.

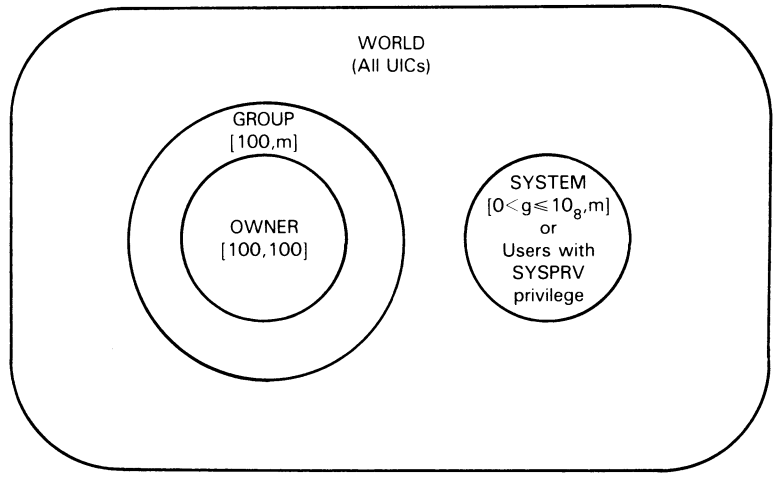
4.2.3 How UIC-Based Protection Controls Access

As indicated in Section 4.1, when a user attempts to gain access to an object (such as a file or volume), in almost all cases the system checks the UIC-based protection code. This involves comparing the user's UIC to the owner UIC of the object. (An exception occurs when there is an ACL on the object that grants access immediately to the requesting user.) Users attempting access to system objects (such as files) always fall into one or more of the following categories:

SYSTEM	One of the following: All users who have system privilege (SYSPRV). Users with low group numbers, usually from 1 through 10 (octal). The exact range of system group numbers is determined by the system manager (with the SYSGEN parameter MAXSYSGROUP) when the system is generated and may range as high as 37776 (octal). These group numbers are generally for system managers, security managers, system programmers, and operators. Users with the user privilege GRPPRV whose UIC group matches the group of the object's owner. For files on disk volumes, users whose UIC matches the owner UIC of the volume on which the file is located.
OWNER	The user with the same UIC as the user who created and therefore owns the object.
GROUP	All users, including the owner, who have the same group number in their UICs as the object's owner.
WORLD	All users, including those in the first three categories.

Figure 4–1 illustrates the relationships of these categories to each other.

Figure 4–1 Illustrating User Categories with a UIC of [100,100]



g = Group Number
m = Member Number

NOTE: THE SYSTEM MANAGER CAN EXTEND THE SYSTEM GROUP NUMBER LIMIT TO 37776_8

ZK-778-82

Through the protection code, each category of user can be allowed or denied any of the following types of access:

- READ
- WRITE
- EXECUTE
- DELETE

CONTROL access is a fifth type of access that can be specified in an ACL and is automatically granted to certain user categories when UIC-based protection is evaluated. CONTROL access grants the accessor all the privileges of the object's actual owner. For example, the user who acquires CONTROL access can change the protection and file characteristics, just as the owner could. Thus, users in the SYSTEM or OWNER categories always have CONTROL access, while users in the GROUP or WORLD categories never receive CONTROL access. The previous list omits CONTROL access since it is never specified in the standard UIC-based protection code.

The significance conveyed by the access types READ, WRITE, EXECUTE, DELETE, and CONTROL varies depending on the situation where they apply. For example, EXECUTE access permits different operations depending on whether it is granted for general file access, directory file access, or volume access. The sections that describe each of these objects also detail the abilities that each access type allows.

The protection code describes the categories of users who have access to an object and the type of access that each category has. For example, the protection code in the next example specifies that users in the SYSTEM and OWNER categories have READ, WRITE, EXECUTE, and DELETE access:

```
SYSTEM:RWED, OWNER:RWED, GROUP:RE, WORLD:RE
```

In this example, users in the GROUP and WORLD categories have only READ and EXECUTE access.

4.2.4 Protection Code Syntax

The following syntax rules apply to protection codes:

- When you specify a protection code, you must abbreviate access types to one character: R, W, E, or D. User categories can be entered in full or truncated to any number of characters. Separate each user category from its access types with a colon. If you specify more than one user category, separate the categories with commas, and enclose the entire code in parentheses. The following DCL command sets the protection code:

  ```
  $ SET PROTECTION=(S:RWED,OWN:RWED,GROUP:R,W:R) DATAFILE.DAT
  ```

- You can specify user categories and access types in any order. If you omit an access type for a user category, that category of user is denied that type of access. If you want to deny all access to a user category, specify the user category, but do not list any access types. Omit the colon after the user category when you are denying access to a category of users. For example, the following DCL command sets the protection code to deny WRITE and DELETE access to the GROUP category and denies all access to the WORLD category:

  ```
  $ SET PROTECTION=(S:RWED,O:RWED,G:RE,W) DATAFILE.DAT
  ```

- When you omit a user category from a protection code, the current access allowed that category of user remains unchanged. For example, assume the protection code in the immediately preceding example is in effect. The following DCL command resets the protection on the file DATAFILE.DAT so that the SYSTEM and OWNER categories of users can no longer DELETE the file, but the GROUP category will still gain READ and EXECUTE access. The WORLD category will still be denied any access:

  ```
  $ SET PROTECTION=(S:RWE,O:RWE) DATAFILE.DAT
  ```

- When you set the protection for magnetic tape volumes, the SYSTEM and OWNER categories always have access, regardless of the specification in the protection code.

4.2.5 How Privileges Affect Protection

The VMS system features privileges that system and security managers can assign to users when they create or modify user accounts. Four system privileges affect user access, regardless of the access dictated by either an ACL for the object or the access granted through matching the user's category in the protection code. Following are these four privileges:

- SYSPRV
- GRPPRV
- READALL
- BYPASS

The following list explains these privileges:

SYSPRV — A user with SYSPRV privilege receives the access accorded to users in the SYSTEM category.

GRPPRV — A user with GRPPRV privilege whose UIC group matches the group of the owner of the object receives the same access accorded to users in the SYSTEM category. Thus, the user with GRPPRV privilege is able to manage a group's files.

BYPASS — A user with BYPASS privilege receives all types of access to the object, regardless of its protection.

READALL — A user with READALL privilege receives READ and CONTROL access to the object, even if that access is denied by the ACL or UIC-based protection. In addition, the user may receive any other access granted through the protection code.

When you define ACLs or UIC protection codes for your objects, remember that users with amplified privileges are entitled to special access to objects throughout the system. For example, there is no way to stop a user with the BYPASS privilege from accessing your files. Protection of your objects depends on the judgment of your security manager in granting these privileges.

4.2.6 How the System Interprets a Protection Code

To determine access to an object (such as a file or a device), the system uses the object's protection code for each user category. The system checks user categories in the following sequence:

1. OWNER
2. WORLD
3. GROUP
4. SYSTEM

You can access a system object as soon as the system finds a user category that you fit into that gives you the access you have requested.

To deny access to a user category, you must deny access to all the outermost categories. For example, the following protection code appears to deny DELETE access to the OWNER category:

```
SYSTEM:RWED, OWNER:RW, GROUP:RW, WORLD:RWED
```

However, the owner of the file can still delete the file. Although DELETE access is denied through the OWNER category, the system continues checking the remaining categories for permission to grant access. Because the owner also fits in the WORLD category (which applies to all users) and the WORLD category is permitted DELETE access, the system grants DELETE access to the owner.

4.2.7 How the System Interprets Object Access Types

Depending on the object, VMS applies different meanings to the access types READ, WRITE, EXECUTE, DELETE, and CONTROL. This section describes several objects and the different interpretations of access types for each.

4.2.7.1 Disk Files

Each file on a disk has its own protection code. When applied to files, access types have the following meanings:

READ	The right to examine (read), print, or copy the file
WRITE	The right to write to or modify the file
EXECUTE	The right to execute a file that contains an executable program image or DCL command procedure
DELETE	The right to delete the file
CONTROL	The right to change the protection and file characteristics of the file

Note that READ access also implies EXECUTE access. WRITE access permits a user to change the contents of a file, even though the file cannot be deleted from the directory without DELETE access.

Note: To open a file for writing, you must have both READ and WRITE access. The VMS operating system does not support write-only files.

4.2.7.2 Directory Files

Each directory file (easily recognized as a file with the file type DIR) has a protection associated with it. This directory protection can override the protection of individual files in the directory. When applied to directories, access types have the following meanings:

READ	The right to examine (read) or list the directory file
WRITE	The right to write to or modify the directory file
EXECUTE	The right to look up files in the directory if you explicitly specify the file name
DELETE	The right to delete the directory file
CONTROL	The right to change the protection and file characteristics of the directory file

Note that READ access implies EXECUTE access.

If you have READ access to a directory file, you can display the contents of the directory file with the DCL command DIRECTORY. You can use wildcards (explicitly or implicitly). For example, if you have READ access to the directory [MALCOLM], you can obtain a listing of all files contained in the [MALCOLM] directory by entering the following command:

```
$ DIRECTORY [MALCOLM]*.*;*
```

You can access any file stored in the directory unless the protection on that file denies you access. However, if a directory denies you READ access, you cannot look up or access even those files in the directory that permit access to users in your group. The Files–11 structure (the file structure used by the VMS operating system) is not hierarchical, and it is possible to write a program to access files without using the directory in which they are listed. Therefore, to guarantee protection, protect individual files as well as directories.

WRITE access allows you to write to the directory file. You must have both READ and WRITE access to a directory in order to create files in that directory, to rename files, or to perform any file operation that involves changes to the directory file.

EXECUTE access allows you to use the DIRECTORY command to look up files that you can identify by name. In addition, you can access files in the directory not protected from users in your category as long as you do not perform an operation that modifies the directory file. You cannot list all the entries in the directory using wildcards.

In the next example, if you have EXECUTE access to the [MALCOLM] directory and you enter a DIRECTORY command without naming any files, the system does not list the files and responds with the following error message:

```
%DIRECT-E-OPENIN, error opening disk:[MALCOLM]*.*;* as input
-RMS-E-PRV, insufficient privilege or file protection violation
```

However, if you know that the file DATAFILE.DAT exists in the [MALCOLM] directory, you can enter a more specific command, as in the following example:

```
$ DIRECTORY [MALCOLM]DATAFILE.DAT;0
```

The system lists the corresponding directory information. EXECUTE access provides some, but not all, of the operations that READ provides.

DELETE access allows you to delete a directory file. You must delete all files from a directory before you can delete the directory file. When you create a directory file with the CREATE/DIRECTORY command, you do not, by default, receive DELETE access. If you want to delete a directory file, you must use the SET PROTECTION command to explicitly assign DELETE access to the OWNER category, as follows:

```
$ SET PROTECTION=OWNER:D TEST.DIR;1
$ DELETE TEST.DIR;1
```

4.2.7.3 Volumes

When applied to volumes, access types have the following meanings:

READ	The right to examine, print, or copy files on a volume
WRITE	The right to modify or to write existing files on a volume
EXECUTE	The right to create files on the volume and to write into them
DELETE	The right to delete files on the volume
CONTROL	The right to change the protection and ownership of the volume

Note that READ access on volumes limits the access to read only.

Note: EXECUTE and DELETE access are not valid for magnetic tapes. Granting a category of users WRITE access to a tape volume automatically permits them to have READ access to the volume.

4.2.7.4 Global Sections

When applied to global sections, access types have the following meanings:

READ	The right to map the section for read access
WRITE	The right to map the section for write access
EXECUTE	The right to map the section for execute access (available only to privileged software)
CONTROL	The right to change the access control list (applies only to PFN and page file global sections)

4.2.7.5 Devices

When applied to devices, access types have the following meanings:

READ	The right to issue read requests to the device
WRITE	The right to issue write requests to the device
CONTROL	The right to change the device ACL

4.2.7.6 Logical Name Tables

When applied to logical name tables, access types have the following meanings:

READ	The right to look up logical names in the table
WRITE	The right to create and delete logical names in the table
DELETE	The right to delete the table
CONTROL	The right to change the logical name table ACL

4.2.7.7 Queues

Operations that apply to a queue or to specific jobs in a queue are controlled by UIC-based protection in the same way access to other system objects is controlled. The ability to control queue operations through UIC-based protection allows you to restrict the types of jobs and users for a particular queue.

When you initialize a queue, the queue is assigned a default owner UIC of [SYSTEM] and the following default protection mask:

```
SYSTEM:EXECUTE,OWNER:DELETE,GROUP:READ,WORLD:WRITE
```

Jobs are assigned an owner UIC equal to the UIC of the process that submitted the job. Each operation performed on a queue or a job in a queue is checked against the owner UIC, protection of the queue and the job, and the privileges of the requestor.

Operations that apply to jobs are checked against the READ and DELETE protection specified for the queue and the owner UIC of the job. In general, READ access to a job allows a user to see the attributes of a job, and DELETE access allows the user to delete the job.

Operations that apply to queues are checked against the WRITE and EXECUTE protection specified for the queue and the owner UIC of the queue. A user with WRITE access to a queue can submit jobs to that queue. A user with EXECUTE access to a queue can act as the operator for that queue with the ability to affect any jobs in the queue. Users with operator (OPER) privilege have EXECUTE access to all queues. OPER privilege also enables users to establish queues and affect accounting.

The following table summarizes the privileges required for various queue operations:

Command	Privilege Required
START/QUEUE/MANAGER	OPER and SYSNAM
STOP/QUEUE/MANAGER	OPER and SYSNAM
DEFINE/CHARACTERISTIC	OPER
DEFINE/FORM	OPER
DELETE/CHARACTERISTIC	OPER
DELETE/FORM	OPER
DELETE/QUEUE	OPER
INITIALIZE/QUEUE	OPER
SHOW QUEUE	OPER, EXECUTE access to the queue, or READ access to the job
SYNCHRONIZE	WRITE access to the queue and READ access to the job
PRINT	WRITE access to the queue and READ access to the file
SUBMIT	WRITE access to the queue and READ access to the file
ASSIGN/MERGE	OPER or EXECUTE access to the queue
ASSIGN/QUEUE	OPER or EXECUTE access to the queue
DEASSIGN/QUEUE	OPER or EXECUTE access to the queue
SET QUEUE	OPER or EXECUTE access to the queue
START/QUEUE	OPER or EXECUTE access to the queue
STOP/QUEUE	OPER or EXECUTE access to the queue
STOP/QUEUE/NEXT	OPER or EXECUTE access to the queue
STOP/QUEUE/RESET	OPER or EXECUTE access to the queue
DELETE/ENTRY	OPER, EXECUTE access to the queue, or DELETE access to the job
SET QUEUE/ENTRY	OPER, EXECUTE access to the queue, or DELETE access to the job
STOP/QUEUE/ABORT	OPER, EXECUTE access to the queue, or DELETE access to the job
SET RESTART_VALUE	No privilege

4.2.8 Establishing and Changing UIC-Based Protection

This section describes how UIC-based protection is initially established for volumes, files, global sections, devices, logical name tables, and queues. It also describes how you can change the protection of volumes, files, devices, and queues.

4.2.8.1 Volume Protection

VMS determines the UIC-based volume protection when a volume is mounted; the default can be taken from the protection recorded on the volume, or it can be explicitly specified. (See the descriptions of the INITIALIZE and MOUNT commands in the *Guide to Maintaining a VMS System* for more information on setting volume protection when you mount a volume.) To change the protection of a disk volume, use the SET VOLUME command.

Volume protection for magnetic tape volumes differs significantly from disk volume protection. The protection applied to a magnetic tape volume applies equally to all files on the volume. VMS applies only READ and WRITE access restrictions to magnetic tapes; EXECUTE and DELETE access are meaningless. Users in the SYSTEM and OWNER category are always given both READ and WRITE access, regardless of what is specified in the protection code. Protection must be explicitly specified when a volume is initialized, or all users will have READ and WRITE access. If you give WRITE access to the GROUP or WORLD categories, READ access is also allowed. For magnetic tapes, users in the SYSTEM and OWNER categories are always given logical and physical I/O access in addition to READ and WRITE access, regardless of what is specified in the protection code.

To change file protection on a magnetic tape, you must reinitialize the tape.

4.2.8.2 Directory Protection

UIC-based directory file protection pertains only to disk directories and is normally established when the directory is created. At directory creation time, you can specify either a protection code with the /PROTECTION qualifier to the DCL command CREATE/DIRECTORY or permit the protection to default to that of the next higher directory in the tree. If the directory is a top-level directory, the protection is taken from the master file directory (MFD). For example, to create the top-level directory file MONROE.DIR with open access to all but the WORLD category of users, the security manager would enter the following command:

```
$ CREATE/DIRECTORY/PROTECTION=(S:RWED,O:RWED,G:RWED,W) -
_$ BOTANYDISK:[MONROE]
```

Any user with CONTROL access can change the protection on the directory with the DCL command SET PROTECTION. For example, the following command changes the protection for the directory [MONROE] by removing access for the GROUP category:

```
$ SET PROTECTION=(S:RWED,O:RWED,G,W) MONROE.DIR
```

4.2.8.3 File Protection

When you create a new file, the file obtains a UIC-based protection code either from the default protection provided by the directory it will reside in or from the default protection of your process. (These defaults are described in Section 4.5.)

A newly created version of an existing file receives the protection code of the previous version of the file. You can specify a protection code when you create a copy of a file. For example, you can use the /PROTECTION qualifier to define the protection for a file you create with the DCL command COPY, as follows:

```
$ COPY USE1:[PAYDATA]PAYROLL.DAT  PAYSORT.DAT -
_$ /PROTECTION=(SYSTEM:RW,OWNER:RWED,GROUP:RW,WORLD)
```

The previous COPY command copies a file from the device USE1 to your default disk directory. The protection code defines the protection for the newly created file PAYSORT.DAT, as follows:

- Users with system UICs (those in the SYSTEM category) can read and write to the file.
- You (in the OWNER category) have all types of access.
- Other users in your group (those in the GROUP category) can read and write to the file.
- All other users (those in the WORLD category) are permitted no access.

You can also change the protection for one of your existing files with the SET PROTECTION command, as follows:

```
$ SET PROTECTION=(SYSTEM:RWE,OWNER:RWED,GROUP:RE,WORLD) -
_$ PAYSORT.EXE
```

4.2.8.4 Global Section Protection

UIC-based protection on global sections, except those backed by disk files, must be reestablished every time the system is booted. If the global section is backed by a disk file, the section protection is derived from the disk file so that changing the file protection changes the section protection. For page frame number (PFN) and page file global sections, set the protection in the $CRMPSC system service call that creates the section; you cannot change the protection after the section is created.

4.2.8.5 Device Protection

UIC-based protection on devices must be reestablished every time the system is booted. Set device protection with the following command:

```
$ SET PROTECTION=(code)/DEVICE device-name[:]
```

4.2.8.6 Logical Name Table Protection

UIC-based protection on logical name tables must be reestablished every time the system is booted. Set the protection with the protection argument to the $CRELNT system service call or with the following command:

```
$ CREATE/NAME_TABLE/PROTECTION=(code) table-name
```

You cannot change the protection on an existing logical name table.

4.2.8.7 Queue Protection

Set UIC-based protection on a queue with the /PROTECTION qualifier to the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. For example, use the following command to change the protection on the FAST$BATCH queue:

```
$ SET QUEUE/PROTECTION=(code) FAST$BATCH
```

4.3 Access Control Lists (ACLs)

An alternative means of protection offered with the VMS operating system is the access control list (ACL). ACLs consist of access control list entries (ACEs) that grant or deny access to a particular system object. Each ACE specifies a user or group of users and the type of access permitted. ACLs define access more precisely than the default UIC-based protection scheme by allowing you to create groups of users independent of the users' UICs.

ACLs can be placed on the following types of objects:

- Devices
- Files (including directory files)
- Group global sections
- System global sections
- Logical name tables
- Queues

VMS provides a file called a *rights database* that contains a list of special names called *identifiers* as well as a list of the users specified as *holders* of identifiers. The security manager uses the VMS Authorize Utility to maintain the rights database, adding and removing identifiers and holders of identifiers as necessary. By allowing groups of users to hold identifiers, the manager has created a group designation that differs from the one used with the user's UIC. This alternative method of grouping is more finely tailored to the uses the holders of the identifier are expected to make of the objects. This method also permits each user to be a member of multiple overlapping groups.

Each time you log in, the system creates a *process rights list* for you containing a list of the identifiers in the rights database associated with your process. When you attempt to access objects protected with ACLs, the system searches the object's ACL for an identifier granting access that matches one of the identifiers in your process rights list.

The following sections describe the relationship between ACLs and identifiers in more detail.

4.3.1 ACLs, Identifiers, and the Reference Monitor

The reference monitor model specifies an authorization database, which describes all access authorizations in the system for all subjects and all objects. This database is often represented as an *access matrix*, listing subjects on one axis and objects on the other. Each crosspoint in the matrix thus represents the access that one subject has to one object.

Figure 4–2 provides an example of an access matrix:

Figure 4–2 Example of an Access Matrix

Objects:	V	W	X	Y	Z
Subjects:					
A		*			*
B		*	*	*	
C		*	*	*	
D	*	*	*	*	
E	*				

In this access matrix, an asterisk (*) is used to denote that the subject has access to that object (different types of access, such as READ and WRITE, are omitted from this example for simplicity). Thus, subjects B, C, and D all have access to objects W, X, and Y. In addition, subject A has access to objects W and Z, subject D to object V, and subject E to object V.

Breaking up the access matrix by rows yields a *capability-based system*, in which each subject carries a list of the objects that it can access. Thus, a capability representation of this access matrix would appear as follows:

A: W, Z
B: W, X, Y
C: W, X, Y
D: V, W, X, Y
E: V

It is also possible to break up the access matrix by columns, listing for each object the subjects that have access to it. This results in an *authority-based system*, or access control lists. The access control list representation appears as follows:

V: D, E
W: A, B, C, D
X: B, C, D
Y: B, C, D
Z: A

The access control list and identifier system that VMS implements combines properties of both the capability- and authority-based systems. The result is an extremely powerful and flexible system capable of representing complex access matrixes in a compact and convenient manner. Consider what happens to the previous example of an access matrix when some of the crosspoints have labels, as in Figure 4–3.

Figure 4–3 Previous Matrix with Labeled Crosspoints

Objects:	V	W	X	Y	Z
Subjects:					
A		*			*
B		Q	Q	Q	
C		Q	Q	Q	
D	P	Q	Q	Q	
E	P				

Some labeled crosspoints can be grouped and treated as a single entity. Thus, the points that are labeled Q represent the access that subjects B, C, and D have to objects W, X, and Y. All the Q points can be considered as a single area of interest. VMS provides the concept of identifiers to take practical advantage of this grouping of areas of interest.

You can define identifiers to represent the two groups of access, P and Q, in the example matrix. Note that two of the crosspoints in the example remain unlabeled. VMS identifiers can also represent individual subjects, and thus allow the traditional access control list facility.

To represent the access matrix, VMS uses two structures, one for each dimension. The system rights database represents the rows of the access matrix, and thus corresponds to the capability model. For this example, you would need the following rights database:

B: Q
C: Q
D: P, Q
E: P

Access control lists on the protected objects represent the columns of the access matrix. For this example, you would need the following access control lists:

V: P
W: A, Q
X: Q
Y: Q
Z: A

Note that the VMS structures required to represent the access matrix are simpler than either the traditional capability or authority model and require fewer terms in total. In the example, the difference is slight. However, complexity of the access matrix increases with the square of its size.

4.3.2 Creating and Maintaining ACLs

Use the VMS ACL Editor to create and edit an ACL on a specific object. You can also use the DCL command SET ACL to manipulate (add, delete, or copy) entire ACLs or individual ACEs on more than one object at a time. For information on the ACL editor, see the *VMS Access Control List Editor Manual.*

The following DCL commands can be used to display ACLs:

- SHOW ACL
- DIRECTORY/ACL
- DIRECTORY/SECURITY
- DIRECTORY/FULL

In general, and in the examples throughout this guide, you will find the DCL commands SET ACL and SHOW ACL sufficient for creating and displaying most ACLs, although the ACL editor is an important utility for more extensive ACL work. For more information on any of these DCL commands, see the individual command descriptions in the *VMS DCL Dictionary.*

You can establish ACLs for various system objects: files, directory files, global sections, devices, queues, and system logical name tables. Be aware of the special considerations described in the following sections when creating ACLs on objects other than files.

4.3.2.1 Global Sections

You must reestablish ACLs on global sections (except those backed by disk files) every time the system is booted because they are not saved.

The ACL on a global section backed by a file is the ACL of the file. Changing the file's ACL causes a corresponding change in the global section's ACL. You cannot change the ACL on the global section.

You can establish ACLs on both system and group global sections. Note, however, that if you attempt to access a group global section outside your UIC group, the operating system denies access and does not consider the ACL. ACLs on PFN and page file global sections can be set up and modified with the ACL editor or with the following commands:

```
$ SET ACL/OBJECT_TYPE=SYSTEM_GLOBAL_SECTION section_name
$ SET ACL/OBJECT_TYPE=GROUP_GLOBAL_SECTION section_name
```

4.3.2.2 Devices

You must reestablish ACLs on devices every time the system is booted because they are not saved. ACLs on devices are set up and modified with the ACL editor or with the following command:

```
$ SET ACL/OBJECT_TYPE=DEVICE device-name
```

4.3.2.3 Logical Name Tables

You must reestablish ACLs on logical name tables every time the system is booted because they are not saved. ACLs can be established for system logical name tables but not process logical name tables. ACLs on system logical name tables are set up and modified with the ACL editor or with the following command:

```
$ SET ACL/OBJECT_TYPE=LOGICAL_NAME_TABLE table-name
```

4.3.2.4 Queues

ACLs on batch and device (printer, server, and terminal) queues are saved in the queue file (JBCSYSQUE.DAT in the SYS$SYSTEM directory) and do not need to be reestablished every time the system is booted.

Set up or modify ACLs on queues with the ACL editor or with the following command:

```
$ SET ACL/OBJECT_TYPE=QUEUE queue-name
```

To deny access to a queue for specific users or groups of users, set up an ACL on the queue denying users all access to the queue, as shown in the following example:

```
$ SET ACL/OBJECT_TYPE=QUEUE/ACL=((IDENTIFIER=JONES,ACCESS=NONE), -
_$ (IDENTIFIER=[DOC,*],ACCESS=NONE)) LN03$PRINT
```

All users except JONES and users in the DOC UIC group can submit jobs to the LN03$PRINT queue.

To limit access to a queue, remove world WRITE access to the queue, and set up an ACL specifying access for all users permitted to submit jobs to the queue. In the following example, only holders of the general identifier PROJECTX can submit jobs to the LN03$PRINT queues. An additional ACE is granted allowing user PIAZZA operator (OPER) privileges to the queue. (If PIAZZA also holds the PROJECTX identifier, this ACE must appear first in the ACL.)

```
$ SET QUEUE/PROTECTION=(W:) LN03$PRINT
$ SET ACL/OBJECT_TYPE=QUEUE/ACL=((IDENTIFIER=PROJECTX,ACCESS=WRITE), -
_$ (IDENTIFIER=PIAZZA,ACCESS=READ+WRITE+EXECUTE+DELETE)) LN03$PRINT
```

Note: If you place an ACL on a generic queue, place an identical ACL on all associated execution queues.

4.3.3 Identifiers

Identifiers in an ACL specify the users who are allowed or denied access to an object. Following are the three types of identifiers:

- *UIC identifiers*—depend on the user identification codes (UICs) that uniquely identify each user on the system. Typically the UIC identifiers are presented in numeric or abbreviated alphanumeric format. For example, a UIC identifier might adopt the numeric format of the UIC, such as [306,210], or just the member name from the alphanumeric format UIC, such as JONES, where the full alphanumeric UIC is [GROUP1,JONES].
- *General identifiers*—defined by the security manager in the system rights database to identify groups of users on the system. For example, TERM3BIO, WARD5WORKERS, DATAENTRY, and RESERVDESK would identify the third term biology students, the campaign workers for Ward 5, the data entry personnel, or the people who handle the reservations desk, respectively.
- *System-defined identifiers*—describe certain types of users based on their use of the system. For example, BATCH, NETWORK, DIALUP, INTERACTIVE, LOCAL, and REMOTE would correspond directly to the descriptions in Section 3.1.1 of the type of login the user executed.

When you log in, the identifiers you hold in the rights database (including your UIC and your system-defined identifiers) are copied into a rights list that is part of your current process. The rights list is the structure that VMS uses to perform all protection checks. Additional identifiers may appear in your rights list; they were put there either by VMS Login software or by software specific to your installation. These identifiers represent qualifications about your login and the state of the system.

4.3.3.1 UIC Identifiers

UIC identifiers conform to the specifications for UICs as presented in the *VMS DCL Concepts Manual*. While the most common types of UIC identifiers are either numeric format UICs or user names, full alphanumeric UICs or UICs in hexadecimal format are accepted as UIC identifiers. Thus, you might see the following UIC identifiers:

```
[PROGRAMMERS,J_JONES]     {alphanumeric format UIC}
J_JONES                   {username from alphanumeric format UIC}
[341,311]                 {numeric format UIC}
%X08001006                {hexadecimal format UIC}
```

Each of these formats uniquely identifies a user.

4.3.3.2 General Identifiers

A general identifier, defined in the system rights database, is an alphanumeric string of 1 through 31 characters that must contain at least one alphabetic character. It can include the characters A through Z, dollar signs ($), underscores (_), and the numbers 0 through 9.

The security manager uses the Authorize Utility (AUTHORIZE) to create and assign general identifiers and UICs to system users.

4.3.3.3 System-Defined Identifiers

System-defined identifiers are automatically defined by the system when the rights database is created at system installation time. The following identifiers, which correspond directly to the login classes that Section 3.1.1 describes, are system-defined:

BATCH	All access attempts made by batch jobs
NETWORK	All access attempts made by DECnet tasks
INTERACTIVE	All access attempts made by interactive processes
LOCAL	All access attempts made by users logged in at local terminals
DIALUP	All access attempts made by users logged in at dialup terminals
REMOTE	All access attempts made by users logged in through a network

In addition, a system node identifier of the form SYS$NODE_*node_name* is created by the site-independent startup procedure (STARTUP.COM in SYS$SYSTEM).

A user automatically becomes a holder of one or more of these identifiers during login. The VMS Login software adds the appropriate identifiers to the process rights list.

4.3.4 Access Control List Entries

The security manager first creates identifiers in the rights database and assigns users as holders of the identifiers, then defines which access to grant or deny holders of the identifier for each object requiring this level of protection. Because several identifiers may be required to represent access needs for each object, it is typical to create a list of multiple entries. Each entry defines the access rights to be granted or denied the holders of the identifier named in that entry. This list is the Access Control List, or ACL. Each entry in this list is called an access control list entry (ACE).

Like the defaults for UIC-based protection, security managers can set up default ACLs. As a result, some users may be unaware that their files have ACLs and may never change ACLs themselves. Other users are actively involved in creating and maintaining their own ACLs.

To summarize, ACLs can be created by the system by default, by the security manager for specific objects, and by users to protect their own files. Users can create ACLs only for objects they own or to which they have the same access as the object owner.

An ACL consists of ACEs that grant or deny access to a particular system object, such as a file, directory, or device. Because ACLs can define access more selectively than UIC-based protection, ACLs allow users to fine tune the action taken when access is requested for an object. Typically, you use ACLs to provide users from several UIC groups access to a system object without having to grant WORLD access to the object. ACLs can perform other functions, such as directing security alarms to be set off when access to an object succeeds or fails.

When the system receives a request for access to an object that has an ACL, the system searches each entry in the ACL sequentially for the first match. It stops searching at the first match. If another match exists further down in the ACL, it has no effect. Thus, ACEs that identify specific users should appear in the ACL before ACEs that identify broader classes of users, as follows:

```
(IDENTIFIER=WILLIAMS,ACCESS=READ+EXECUTE)
(IDENTIFIER=CS101,ACCESS=NONE)
```

Assume that user WILLIAMS holds the CS101 identifier. In the previous example, WILLIAMS is granted READ and EXECUTE access to the object. If the ACEs were switched, user WILLIAMS may be denied access to the object.

The use of ACLs is optional. Although the use of ACLs can enhance the security of system objects in any installation through a more detailed definition of who is allowed what kind of access, user time must be spent in creating and maintaining the ACLs, and processor time is required to perform the functions that ACLs mandate.

Each ACL consists of one or more ACEs. There is no limit to the number of ACEs that an ACL can contain or to the number of characters in an ACE; however, very long ACLs increase the amount of time necessary to gain access to an object.

The type of access protection needed determines the type of ACE used in a given situation. Following are three types of ACEs involved with security:

- *Identifier ACE*—Controls the type of access allowed to a particular user or group of users.

- *Default protection ACE*—Defines the default protection for a directory so protection can be propagated to the files and subdirectories created in that directory. (This type of ACE is applicable only to directory files.)
- *Security alarm ACE*—Provides an alarm message when an object is accessed to help alert managers to possible security threats.

The exact format of an ACE depends on its type, but all ACEs are enclosed in parentheses. In general, the format of an ACE is as follows:

```
(type[,options][,access_to_grant])
```

4.3.4.1 Identifier ACE

An identifier ACE controls the types of access allowed to specific users based on user identification. Following is the format for an identifier ACE:

```
(IDENTIFIER=identifier[,options][,access])
```

Specifying Identifiers in Identifier ACEs

The first field in the identifier ACE is the keyword IDENTIFIER followed by one or more identifiers. An identifier can be one of the following:

- User identification code (UIC)
- General, established by the system manager in the system rights database
- System-defined

A UIC can be in either numeric or named UIC format, as described in the *VMS DCL Concepts Manual*.

A general identifier, defined in the system rights database, is an alphanumeric string of 1 through 31 characters that must contain at least one alphabetic character. It can include the characters A through Z, dollar signs ($), underscores (_), and the numbers 0 through 9.

The system manager creates and assigns general identifiers and UICs to system users using the Authorize Utility (AUTHORIZE).

System-defined identifiers are automatically defined by the system when the system manager creates a rights database. The following identifiers are system-defined identifiers:

BATCH	All attempts at access made by batch jobs
NETWORK	All attempts at access made over the DECnet–VAX network
INTERACTIVE	All attempts at access made by interactive processes
LOCAL	All attempts at access made by users logged in at local terminals
DIALUP	All attempts at access made by users logged in at dialup terminals
REMOTE	All attempts at access made by users logged in via a network

Generally, use only one of the six system-defined identifiers at a time. You can use them with other identifiers (UICs and general identifiers). When you specify multiple identifiers, connect them with plus signs (+).

The system takes the access action included in the ACE only for the user who matches all the identifiers. For example, if you wanted to grant read access to user [301,25] running a batch job, you would specify the identifier ACE as follows:

```
(IDENTIFIER=[301,25]+BATCH,ACCESS=READ)
```

Although it is unusual for a number of users to share the same UIC, it is likely that a number of users will share the same general identifier. Users with the same general identifier do not need to be in the same UIC-based group. Furthermore, a single user can be associated with a number of different general identifiers as defined in the rights database. The creator of an ACL has considerable flexibility in selecting sets of users and defining access capabilities for them.

For example, the user identified by the UIC [301,25] is a member of the UIC-based group 301. That user may be the only member of group 301 who is also associated with the general identifier PERSONNEL. An ACE defining a particular type of access for the users associated with the general identifier PERSONNEL grants that type of access to that user, but not to the other members of group 301.

Specifying Options in Identifier ACEs

The options field in an identifier ACE controls whether an ACE is propagated, can be displayed, or can be deleted. This field in an identifier ACE begins with the keyword OPTIONS and takes one or more of the following keywords:

DEFAULT	Indicates that an ACE is to be included in the ACL of any files created within a directory. When the ACE is propagated, the DEFAULT indicator is removed from the ACL of the created file. This option is valid only for directory files. A default ACE does not grant or deny access; it just affects the ACL of new files.
HIDDEN	Indicates that this ACE should only be changed by the application that added it. The ACL editor does not permit modification or deletion. Thus, the ACL editor displays the ACE only to show its relative position within the ACL, not to facilitate editing of the ACE. The DCL DIRECTORY and SHOW ACL commands do not display hidden ACEs.
PROTECTED	Indicates that an ACE will be preserved even when an attempt is made to delete the entire ACL. A protected ACE must be deleted specifically with the ACL editor or by specifying the ACE on the command line of the DCL command SET ACL.
NOPROPAGATE	Indicates that, when copying an ACL from one version of a file to a later version of the same file, the ACE is not copied to the newer version.
NONE	Indicates that no options apply to an ACE. Although you can enter OPTIONS=NONE when you create the ACE, OPTIONS=NONE is not displayed when the ACE is displayed.

Connect multiple options with plus signs (+). If you specify any other options with the NONE option, the other options take precedence.

Identifier ACE for a Directory

The OPTIONS=DEFAULT option of an identifier ACE allows users to define one or more default ACEs for inclusion in the ACLs for files created in a particular directory. A default ACE is supplied for all new files created in that directory; any existing files are not supplied with the default ACE. Thus, if you wanted all files in the directory [MALCOLM] to have an ACE that permitted read and write access to users with the PERSONNEL identifier, you could include the following ACE in the ACL for the file MALCOLM.DIR:

```
(IDENTIFIER=PERSONNEL,OPTIONS=DEFAULT,ACCESS=READ+WRITE)
```

As a result of this ACE, any file created in the [MALCOLM] directory has the following ACE:

```
(IDENTIFIER=PERSONNEL,ACCESS=READ+WRITE)
```

Notice that the DEFAULT option does not appear in the file's ACE. However, any subdirectory created in the MALCOLM directory has the DEFAULT option as part of its ACE so that the default ACE will be propagated throughout the entire directory tree.

Specifying Access in Identifier ACEs

The third field in an identifier ACE specifies what type of access you are allowing the users identified in the first field of the ACE. This field begins with the keyword ACCESS followed by a string of access actions connected by plus signs. The following types of access are allowed in an identifier ACE:

READ	Accessor can read a file, read from a disk, or allocate a device.
WRITE	Accessor can read or write a file.
EXECUTE	Accessor can execute an image file or look up entries in a directory by explicitly specifying file names.
DELETE	Accessor can delete a file.
CONTROL	Accessor has all the privileges of the object's owner.
NONE	Accessor has no access to the object.

Sample Identifier ACEs

The most common type of ACL is one that defines the access to a file for a group of users. In the following ACL example, access to a file is based on the identity of a user. PERSONNEL, SECURITY, and SECRETARIES are general identifiers assigned to appropriate sets of users by the system manager using AUTHORIZE. NETWORK is a system-defined identifier, while [20,*] and [SALES,JONES] are examples of UIC identifiers.

```
(IDENTIFIER=SECURITY,OPTIONS=PROTECTED,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=PERSONNEL,ACCESS=READ+WRITE+EXECUTE+DELETE)
(IDENTIFIER=SECRETARIES,ACCESS=READ+WRITE)
(IDENTIFIER=[20,*],ACCESS=READ)
(IDENTIFIER=NETWORK,ACCESS=NONE)
(IDENTIFIER=[SALES,JONES],ACCESS=NONE)
```

In the preceding example, the ACE providing the greatest amount of file access is listed at the top of the ACL. Any users holding both the SECURITY and PERSONNEL identifiers obtain maximum access rights through the first match, which is the SECURITY identifier. In this example, the user with UIC [SALES,JONES] is prohibited from any access to the file unless that user also happens to have one of the general identifiers (which is an oversight on the part of the creator of the ACL). If the ACL creator wants to be absolutely

certain that the user with UIC [SALES,JONES] could not possibly gain access to the file, the ACE at the bottom of the ACL should be moved to the top.

The order of the ACEs in the example permits a number of users to gain types of file access over the DECnet–VAX network. The users with the identifiers of SECURITY, PERSONNEL, SECRETARIES, and UIC [20,*] can all gain some access over the network, although only those with the identifier SECURITY can gain full access. The fifth ACE prevents all other users from network access. While this might be the intent of the ACL creator, it would be an unfortunate oversight if it were not. Remember that the system searches the ACL sequentially and grants the user only the access specified in the first matching ACE. All subsequent ACEs are ignored.

The first ACE is the only ACE containing an option field (the PROTECTED option). Using this option prevents the first ACE from being deleted unless you have explicitly deleted the ACE with the ACL editor, or you have specified the ACE with the SET ACL/DELETE command.

Identifier ACEs for Other Objects

Create identifier ACEs for other system objects, such as devices, as you create ACEs for files or directories. For example, suppose your company has a special letter-quality printer (TTA8) that is used only for printing checks. As a result, the check forms are always loaded in the printer. This device is never to be used for logins, and no queues are directed to it. Only one user, MGREY, is allowed read and write access to it. The system manager can establish this restriction by setting the protection on the printer with the following command:

```
$ SET PROTECTION=(S,O,G,W)/DEVICE TTA8:
```

The following identifier ACE, applied to the object TTA8, restricts access to the device:

```
(IDENTIFIER=MGREY,ACCESS=READ+WRITE)
```

4.3.4.2 Default Protection ACE

The default protection ACE is used to ensure that one type of UIC-based protection is propagated throughout a directory tree. This type of ACE allows you to specify protection for one directory structure that is different from the default protection applied to other directories. Default protection ACEs can be applied only to directory files.

Following is the format for a default protection ACE:

(DEFAULT_PROTECTION[,options],protection_mask)

This type of ACE is specified by the keyword DEFAULT_PROTECTION. The second field (the options field) in a default protection ACE controls whether an ACE is propagated, can be displayed, or can be deleted. This field in a default protection ACE begins with the keyword OPTIONS and takes one or more of the following keywords:

HIDDEN	Indicates that this ACE can only be changed by the application that added it. The ACL editor does not permit modification or deletion. Thus, the ACL editor displays the ACE only to show its relative position within the ACL, not to facilitate editing of the ACE. The DCL DIRECTORY and SHOW ACL commands will not display hidden ACEs.
PROTECTED	Indicates that an ACE is preserved even when an attempt is made to delete the entire ACL. A protected ACE must be specifically deleted with the ACL editor or by specifying the ACE on the command line of the DCL command SET ACL.
NOPROPAGATE	Indicates that, when copying an ACL from one version of a file to a later version of the same file, the ACE is not propagated.
NONE	Indicates that no options apply to an ACE. Although you might enter OPTIONS=NONE when you create the ACE, OPTIONS=NONE is not displayed when the ACE is displayed.

Connect multiple options with plus signs (+). If you specify any other options with the NONE option, the other options will take precedence.

The protection mask is specified the same as for UIC-based protection, with the user categories—SYSTEM, WORLD, GROUP, and OWNER—and the access categories—READ, WRITE, EXECUTE, and DELETE. See the discussion of UIC-based protection in the *VMS DCL Dictionary* for more information.

The following sample ACE, included in an ACL for the directory MALCOLM, sets up default protection so that any files created in the directory allow system and owner groups read, write, execute, and delete access. Group and world groups are denied access.

```
(DEFAULT_PROTECTION,S:RWED,O:RWED)
```

When you add or change the default protection for a directory, there is no effect on the files already created in the directory. All new files will receive the default protection.

If you want to have the default protection ACE PROTECTED, which saves its ACE if an attempt is made to delete the entire ACL, create the following ACL:

```
(DEFAULT_PROTECTION,OPTIONS=PROTECTED,S:RWEDC,O:RWEDC,G,W)
```

4.3.4.3 Security Alarm ACE

The security alarm ACE allows you to specify that an alarm message be sent to the security operator's terminal if a certain type of access takes place. (The DCL command SET AUDIT enables the security operator's terminal to receive security alarms.)

The security alarm ACE specifies the type of access that you want to protect. When the specified access is violated, an alarm message is sent to security operators.

Although you can create alarm ACEs in an ACL that cause the system to observe the event and take the required action, you should also coordinate protection with your system's security manager (the person who possesses the SECURITY privilege). The security manager is responsible for enabling the alarm feature. Since this feature uses system resources, the security manager might be reluctant to leave it enabled at all times.

Following is the format of a security alarm ACE:

(ALARM_JOURNAL=SECURITY[,options][,access])

This type of ACE is specified by the keywords ALARM_JOURNAL=SECURITY. The second field in a security alarm ACE begins with the keyword OPTIONS, which takes one or more of the following keywords:

DEFAULT	This option is valid only for directory files. Indicates that an ACE is to be included in the ACL of any files created within a directory. When the ACE is propagated, the DEFAULT indicator is removed from the ACL of the created file.
HIDDEN	Indicates that this ACE can only be changed by the application that added it. The ACL editor does not permit modification or deletion. Thus, the ACL editor displays the ACE only to show its relative position within the ACL, not to facilitate editing of the ACE. The DCL DIRECTORY and SHOW ACL commands will not display hidden ACEs.
PROTECTED	Indicates that this ACE is preserved even when an attempt is made to delete the entire ACL. A protected ACE must be explicitly deleted with the ACL editor or by specifying the ACE on the command line of the DCL command SET ACL.
NOPROPAGATE	Indicates that, when copying an ACL from one version of a file to a later version of the same file, the ACE is not propagated.
NONE	Indicates that no options apply to this ACE. Although you enter OPTIONS=NONE when you create the ACE, OPTIONS=NONE is not displayed when the ACE is displayed.

Connect multiple options with plus signs (+). If you specify any other options when specifying NONE, the other options take precedence.

The third field in an alarm ACE controls the type of access that causes the alarm to be sent. Specify any of the following access actions with the ACCESS keyword:

READ	Generates an alarm if an accessor attempts to read the object.
WRITE	Generates an alarm if an accessor attempts to read or write the object.
EXECUTE	Generates an alarm if an accessor attempts to execute the object.
DELETE	Generates an alarm if an accessor attempts to delete the object.
CONTROL	Generates an alarm if an accessor attempts to perform control operations on the object, such as changing the protection on the object.
SUCCESS	Generates an alarm for each successful attempt by an accessor to access the object.
FAILURE	Generates an alarm for each unsuccessful attempt by an accessor to access the object.

Note: For an alarm to have any effect, you must include SUCCESS or FAILURE or both.

4.3.5 Summary of ACLs

The following recommendations will help you manage ACLs:

- Do not assume that specifying ACCESS=NONE for an identifier will absolutely prohibit the holders of the identifier from accessing the object. Frequently, users in either the SYSTEM or OWNER category may still be entitled to whatever access the UIC-based protection affords that category. If the users hold special privileges, they may be granted the access requested through the privilege.
- Watch out for errors in the order that ACEs appear in the ACL. Place the ACEs that deny access to specific users at the top of your ACLs, so that the user will not obtain access by holding another identifier. Sometimes you use wildcards in the UIC-format identifiers to deny access to large groups of users. Such an ACE properly belongs at the bottom of the ACL, not at the top. Place the ACEs that grant the widest access rights immediately before the most restrictive ACEs. This technique ensures that users who hold multiple identifiers do not obtain restricted access rights on the first match when another identifier they hold could grant more generous rights. Remember that a user can only receive the access rights granted through the first matching identifier.
- Do not place ACLs on all objects. This is usually unnecessary even at medium-level security sites. Too many ACLs can cause performance penalties to appear on the system. Instead of using ACLs, group files so that only a few directories need default ACEs that propagate to many or all files.
- Use general identifiers to create practical groups of users to avoid unnecessarily long ACLs.
- Update ACLs when users leave. Always maintain the shortest and most current ACLs. Again, using general identifiers instead of individual users helps alleviate this maintenance problem.

4.4 Establishing and Changing Object Ownership

This section describes how VMS establishes the ownership of resources such as volumes, directories, and files. It includes an explanation of the attributes users may have associated with some of their identifiers and how those attributes can affect the default file ownership. The section also describes the requirements VMS imposes on users before allowing them to change object ownership. The appropriate DCL commands are given.

4.4.1 Understanding the Role of Identifier Attributes

Attributes are identifier characteristics that you can specify when adding identifiers to the rights database or granting identifiers to users. You can specify the following attributes when adding or granting identifiers:

- Resource—allows holders of the identifier to charge disk space to the identifier
- Dynamic—allows holders of the identifier to remove the identifier from the process rights list using the DCL command SET RIGHTS_LIST

4.4.1.1 Resource Attribute

Consumption of disk space is generally charged to the creator of each file by subtracting the disk space from the file owner's disk quota. System and security managers may prefer to track the use of disk space according to logical groups of users (such as departments or projects) rather than individual users. General identifiers can specify these groups. Thus, when general identifiers own directories, disk space used by files created in the directories is charged to the identifier.

To allow file space to be owned by and charged to an identifier, specify the resource attribute when adding the identifier to the rights database using AUTHORIZE, as shown in the following example:

```
$ SET DEF SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD/IDENTIFIER MGMT101 /ATTRIBUTES=RESOURCE
```

To allow specific holders of the identifier to charge disk space to the identifier, include the resource attribute when granting the identifier, as follows:

```
UAF> GRANT/IDENTIFIER MGMT101/ATTRIBUTES=RESOURCE JONES
```

Not everyone who holds the identifier will also hold the resource attribute associated with that identifier. If you create a file in a directory owned by an identifier, and you do not have the resource attribute for that identifier, the required disk space is subtracted from your disk quota.

4.4.1.2 Dynamic Attribute

Once you grant an identifier to a user, the user usually holds the identifier (in the process rights list) until logged out. However, if you grant the identifier with the dynamic attribute, the user who holds the identifier can use the DCL command SET RIGHTS_LIST to add or remove the identifier or its attributes from the process rights list as needed.

To allow users to modify an identifier, specify the dynamic attribute when adding the identifier to the rights database using AUTHORIZE, as shown in the following example:

```
$ SET DEF SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD/IDENTIFIER MGMT101 /ATTRIBUTES=DYNAMIC
```

To allow specific holders of the identifier to modify the identifier, include the dynamic attribute when granting the identifier, as follows:

```
UAF> GRANT/IDENTIFIER MGMT101/ATTRIBUTES=DYNAMIC JONES
```

User JONES enters the following command to remove the MGMT101 identifier from the process right list:

```
$ SET RIGHTS_LIST/DISABLE MGMT101
```

Users who hold identifiers with the dynamic and resource attributes can also use the SET RIGHTS_LIST command to remove just the resource attribute on the identifier. See the *VMS DCL Dictionary* for a complete description of the SET RIGHTS_LIST command.

Because users may be able to circumvent system security by removing their identifiers, be careful when granting users an identifier with the dynamic attribute. If a user who holds an identifier with the dynamic attribute removes the identifier from the rights list, and holders of the identifier were explicitly

denied access in an ACL, the user may then gain access to the object through another ACE in the ACL.

As a privileged security manager, you can use the SET RIGHTS_LIST command to modify the rights list of any process on the system or to modify identifiers in the system rights list. The following privileges are required:

- Modifying identifiers of processes in your UIC group requires CMKRNL and GROUP privileges.
- Modifying identifiers of any process on the system requires CMKRNL and WORLD privileges.
- Modifying the system rights list requires CMKRNL and SYSNAM privilege.

See the *VMS DCL Dictionary* for a complete description of the SET RIGHTS_LIST command.

4.4.2 Defining the Conditions That Convey Ownership Privileges

Descriptions throughout this section refer to the conditions that qualify a user to set or change ownership of an object. The user who meets these conditions has *ownership privileges* for the object. That is, the user may or may not be the recorded owner for the object, but by virtue of possessing a VMS privilege or holding an identifier with the resource attribute, the user may be entitled to set or change the ownership.

The general conditions that would convey ownership privileges for actions involving directories and files are identical. (The conditions for conveying ownership privileges to volumes differ. See Section 4.4.3.)

Users need to satisfy only one of these conditions to receive ownership privileges:

- Hold the resource attribute to the identifier that owns the file
- Qualify as members of the SYSTEM user category, hold SYSPRV or BYPASS privilege, or hold a UIC that matches that of the owner of the volume containing the file or directory
- Hold the GRPPRV privilege while also holding a UIC in the same group as the object owner

If a user wants to set or change the ownership of an object, the user must have ownership privileges for the identifier that owns the object. However, if the user wants to change the ownership of an existing object, the user must have ownership privileges for **both** old and new owner identifiers.

4.4.3 Establishing and Changing Volume Ownership

The ownership of a volume is established when you initialize the volume with the DCL command INTITIALIZE. Unless you specify an owner with the qualifier /OWNER_UIC, the owner of the volume defaults to your current process.

Display volume ownership with the DCL command SHOW DEVICES/FULL.

To change the owner of a volume, enter the DCL command SET VOLUME/OWNER_UIC. Only users who can match one of the following criteria can change the ownership of the volume:

- Qualify as a member of the SYSTEM category of users
- Hold the SYSPRV privilege
- Own the volume

4.4.4 Establishing and Changing Directory Ownership

Ownership of directories is usually set when the directory is created with the DCL command CREATE/DIRECTORY. A user with ownership privileges (see Section 4.4.2) can specify the owner by using the DCL command CREATE/DIRECTORY/OWNER_UIC. If no owner is explicity specified, VMS assigns a default owner, as follows:

- If the directory name is in alphanumeric or subdirectory format, ownership defaults to the owner UIC of the existing parent directory if the user has ownership privileges to that UIC; if not, the ownership defaults to the UIC of the process issuing the command.
- If the directory name is in UIC format, the ownership defaults to the UIC in the directory name.

Display the directory file owner by entering the DCL command DIRECTORY/OWNER. For example, to check the directory file ownership for all top-level directories, use the following command:

```
$ DIRECTORY/OWNER [000000]*.DIR
```

View the owners of all your subdirectories with the following command:

```
$ DIRECTORY/OWNER [...]*.DIR
```

However, if there are subdirectories for which you are not the owner and you fail the protection check, you will be unable to view owner information

If you have ownership privileges to the directory, you can change the directory file ownership with the DCL command SET FILE/OWNER_UIC. For example, to change the current owner of the subdirectory [MONROE.WEATHER] to [CHEM4,LEONARD], you would enter the following command:

```
$ SET FILE/OWNER_UIC=[CHEM4,LEONARD] [MONROE]WEATHER.DIR
```

The same rules of establishing and changing directory ownership apply when the directory is owned by a general identifier.

4.4.5 Establishing and Changing File Ownership

The VMS operating system selects a default owner for a file in the following order:

1. An attempt is made to propagate the ownership from a previous version of a file. This succeeds only if the user has ownership privileges to the previous version. (Ownership privileges are defined in Section 4.4.2.)
2. If the attempt to propagate from the previous version fails (either because there is no previous version or the file creator lacks ownership privileges to the previous version), then an attempt is made to propagate ownership from the parent directory. This succeeds only if the user has ownership privileges to the parent directory. (Ownership privileges are defined in Section 4.4.2.)
3. If the attempt to propagate from the parent directory fails, the owner of the file is the same as the creator of the file.

Users with ownership privileges can specify an alternative file owner with the CREATE/OWNER_UIC command.

You can display the current file owner with the DCL command DIRECTORY/OWNER. However, if there are files in the directory for which you are not the owner and you fail the protection check, you will be unable to view the owner information.

Only those users who have ownership privileges (see Section 4.4.2) can change file ownership. If you have ownership privileges, you can change file ownership with the DCL command SET FILE/OWNER_UIC.

4.5 Propagation of Protection Defaults

This section describes ACL protection defaults for directories and files. It also explains propagation of both UIC-based and ACL-based protection.

4.5.1 Default Directory File Protection

Directory file protection establishes protection and provides default values that can be applied to files added to the directory when they lack specifications of their own. Both UIC-based protection and ACL protection can be placed on directory files.

4.5.1.1 Default UIC-Based Directory File Protection

Directory files receive their file protection codes at creation time. The directory file protection code of a subdirectory defaults to that of its next higher level directory. The default protection code of a top-level directory comes from the volume master file directory. The creator of the directory always has the option of specifying a protection code with the DCL command CREATE DIRECTORY/PROTECTION.

You can change the directory file protection by using either of the DCL commands SET PROTECTION or SET FILE/PROTECTION.

4.5.1.2 Default ACL Protection

Frequently it is efficient to set up one ACL to apply to every new subdirectory file to be created in the directory tree. By default, a subdirectory inherits the ACL of its parent. Use the NOPROPAGATE option in the ACL to specify which ACEs are not to be copied to newly created subdirectories.

With this technique, you can also provide default ACL entries for propagation into ACLs for the files kept in the directory. (Use the DEFAULT option to specify an ACE to be copied to all files created in the directory.)

4.5.2 Default File Protecton

All files need UIC-based protection. Some files require ACL protection as well. The following sections describe how to recognize needs for UIC-based and ACL-based default file protection, as well as how to override those defaults.

Note: To adequately protect files, you must also protect the directory in which the files reside.

4.5.2.1 Default UIC-Based Protection

If you do not define a protection code for a file when you create the file, the system applies a default protection code.

When you create a file, protection is determined sequentially, as follows:

1. If the file is a new version of an existing file, the new file inherits the protection of the existing version.
2. If no previous version exists, and the directory where the file is to be stored has an associated ACL that includes a DEFAULT_PROTECTION ACE, the protection specified by that ACE is used.
3. If the directory does not have a DEFAULT_PROTECTION ACE, the default process protection is used. VMS consults the SYSGEN parameter RMS_FILEPROT to establish this value during login. However, the value derived at login may have been overridden explicitly by you (or possibly by your login command procedure) with the DCL command SET PROTECTION/DEFAULT.

   You can determine your current process default protection by entering the DCL command SHOW PROTECTION, as follows:

```
$ SHOW PROTECTION
  SYSTEM=RWED, OWNER=RWED, GROUP=RE, WORLD=NO ACCESS
```

   The system responds by displaying your default protection. In this example, it indicates that users in the SYSTEM and OWNER categories have all types of access, that members of the owner's group (in the GROUP category) have READ and EXECUTE access, and that all other users (in the WORLD category) have no access.

At any time during your terminal session you can change the default process protection applied to files that you create by invoking the DCL command SET PROTECTION/DEFAULT.

To determine the current protection associated with a specific file or files, use the /PROTECTION qualifier with the DCL command DIRECTORY, as follows:

```
$ DIRECTORY/PROTECTION  PERSONNEL.REC

Directory USE1:[CRAMER]

PERSONNEL.REC;5        (RWED,RWED,RW,R)

Total of 1 file.
```

4.5.2.2 Default ACL Protection

When you create a file, it may receive an ACL by default from one of the following sources:

- If the file is a new version of an existing file, the new file inherits the ACL of the existing one, without any ACEs marked with the NOPROPAGATE option.
- If no previous verson of the file exists, VMS checks the directory in which you are creating the file for an ACL. If there is an ACL, all entries in the ACL marked with the DEFAULT option are copied to the new file with the DEFAULT option removed.

In addition, when you create a file whose owner identifier is not your UIC (by explicitly naming a file owner or through the ownership defaulting rules presented in Section 4.4.5), an ACE that grants CONTROL access to your UIC plus the access available to the owner of the file is added to the file's ACL. This feature guarantees that you retain control over a file that you have just created, regardless of other defaults.

You can use wildcards in the SET ACL command to make identical changes to ACLs on a large number of files. For example, if you decide you want to add an ACE to every ACL in your top level directory to grant holders of the identifier SPECIAL both READ and CONTROL access, you would enter the following command:

```
$ SET ACL/ACL=-
_$ (IDENTIFIER=SPECIAL,ACCESS=READ+CONTROL) *.*;*
```

To avoid recreating ACLs, find an existing ACL that you want to copy. Use the DCL command SET ACL/LIKE to place a copy of an existing ACL on one of your files. For example, to place the ACL that currently exists on the file NEWTERM.MEM onto the file FALLTERM.MEM, enter the following command:

```
$ SET ACL/LIKE=NEWTERM.MEM FALLTERM.MEM
```

Since the SET/ACL/LIKE command will also accept wildcard specifications, you are able to copy the final edited version of the ACL to many files.

4.6 Summary of File Protection Evaluation

Figure 4–4 charts the sequence of procedures that VMS follows when evaluating an access request and shows how the three controlling components (ACLs, protection codes, and privileges) interact.

Figure 4–4 Flowchart of Access Request Evaluation

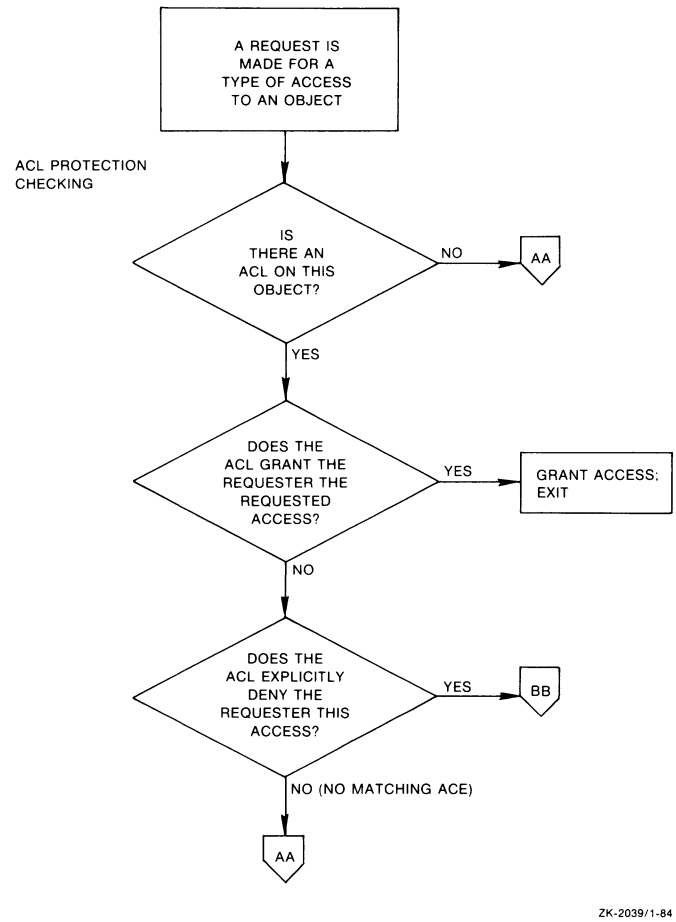


Figure 4–4 Cont'd. on next page

Figure 4–4 (Cont.) Flowchart of Access Request Evaluation

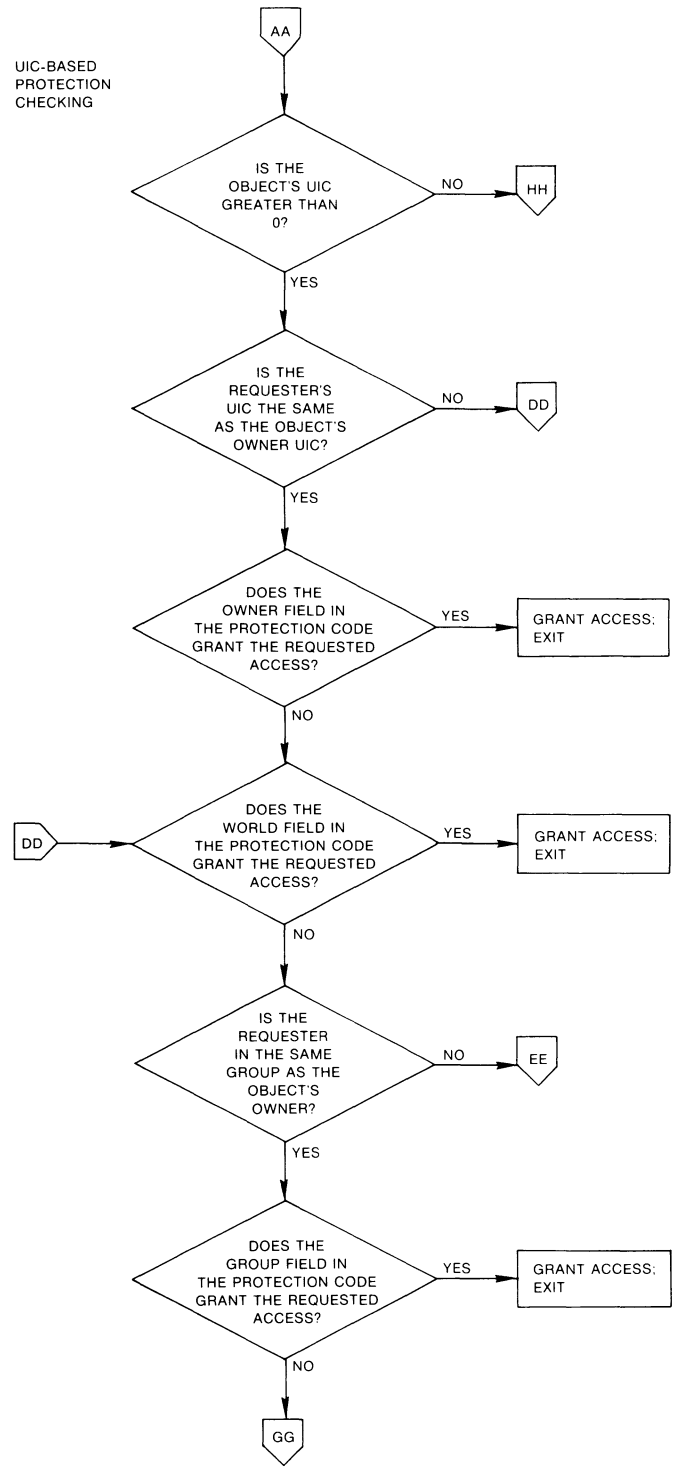


Figure 4–4 Cont'd. on next page

Figure 4–4 (Cont.) Flowchart of Access Request Evaluation

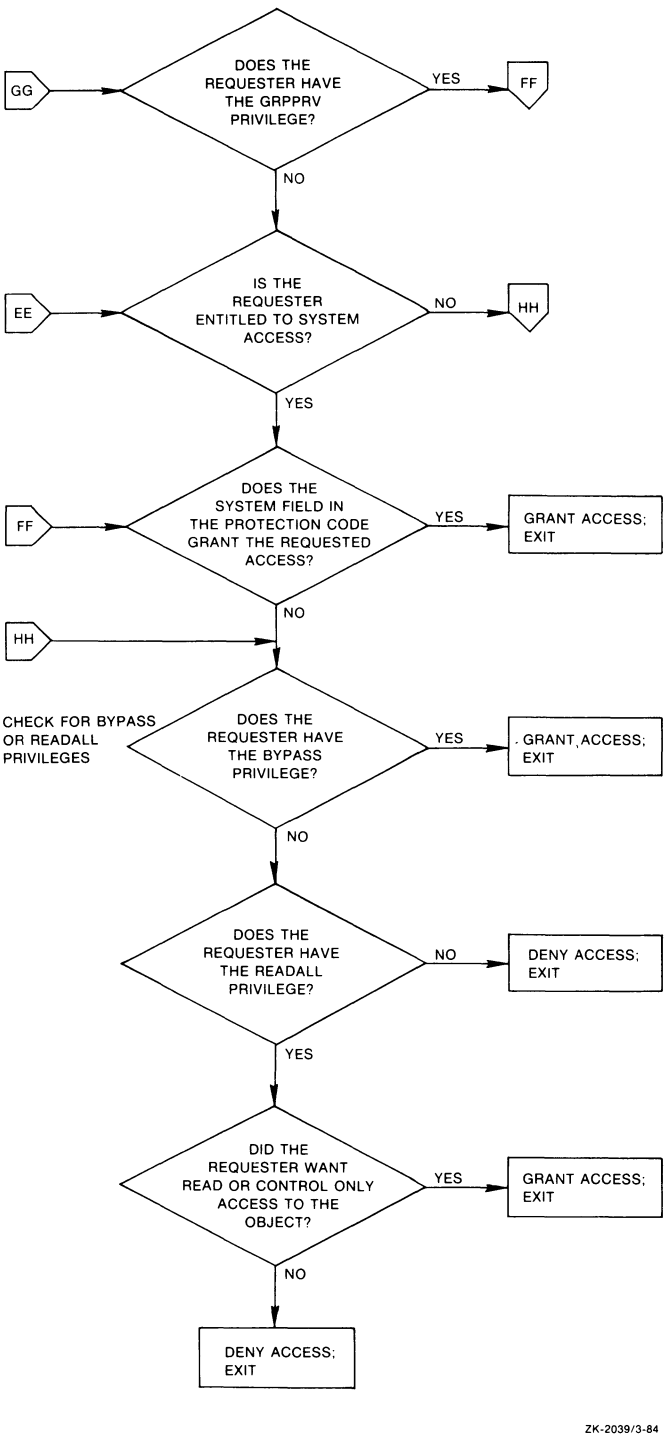
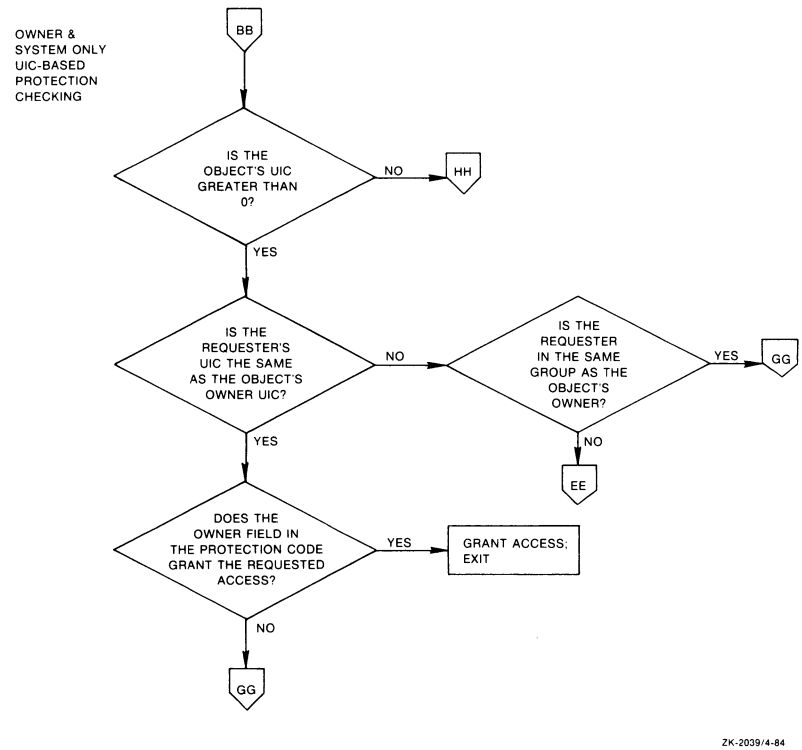


Figure 4–4 Cont'd. on next page

Figure 4–4 (Cont.) Flowchart of Access Request Evaluation



4.7 Protecting Purged or Deleted Data from Disk Scavenging

Because data exists on a disk until it is overwritten, it is necessary to protect deleted or purged file information from *disk scavenging*. Although a deleted or purged file header record denies normal access, a disk scavenger can use special programs and equipment to read those files. With advanced equipment, it is also possible to read the faint residual magnetic impressions of overwritten files. Therefore, sites with medium- to high-level security requirements must adequately protect against disk scavenging.

The VMS operating system offers the following disk scavenging protection: *erasure patterns* and *highwater marking*.

4.7.1 Erasure Patterns

An erasure pattern is a repeated sequence of bits written over a file when the file is deleted or purged. The system manager can automatically apply the erasure pattern to each deleted file by specifying the /ERASE qualifier when the volume is initialized, as follows:

```
$ INITIALIZE/ERASE device-name[:]  volume-label
```

If the volume is mounted, the system manager can apply the erasure pattern with the following command:

```
$ SET VOLUME/ERASE_ON_DELETE device-spec[:]
```

Alternatively, users may be requested to specify the erasure pattern on a file-by-file basis by using the /ERASE qualifier when entering the DCL commands SET FILE, DELETE, and PURGE.

4.7.2 Highwater Marking

Highwater marking keeps users from reading file space beyond the areas where they have been permitted to write. The outer limit of written space on the file is that file's highwater mark. This technique prevents users from scavenging portions of the disk for information that they did not actually write.

The VMS operating system implements highwater marking on sequential, exclusively-accessed files, such as files output from various text editors, compilers, and linkers; that is, most files a process writes. For indexed and shared sequential files, a technique known as *erase-on-allocate* is used in place of true highwater marking. Erase-on-allocate means that blocks of disk space are erased as they are allocated to the user. By default, highwater marking is enabled when the volume is initialized. The system or security manager can disable highwater marking for a specific volume by using the DCL command SET VOLUME/NOHIGHWATER.

4.8 User Auditing

Although it is the security manager's job to monitor the system for possible break-in attempts, users can apply several techniques to assist the security manager in auditing access to their account and files.

This section describes user auditing techniques.

4.8.1 Noting Your Last Login Time

In your UAF record, VMS maintains information about last logins into your account. Your security manager decides whether the system should display this information at login time. Sites with medium to high security requirements frequently display this information and ask users to check it for unusual or unexplained successful logins and unexplained failed logins.

If there is a report of interactive or noninteractive login at a time when you were not logged in, report it promptly to your security manager and change your password immediately. The security manager can investigate further using the system accounting and audit logs.

If you receive a login failure message and cannot account for the failure, it is likely that someone has been trying to access your account unsuccessfully. Check your password to ensure that it adheres to all recommendations for password security described in Section 3.1.3.10. If not, change your password immediately.

If you expect to see a login failure message and it does not appear, or if the count of failures is too low, change your password immediately. Report either of these indications of login failure problems to your security manager.

4.8.2 Tools for Detecting System Abuse

The VMS operating system provides the security manager with many tools to assist in detecting system abuses. Among these tools are security alarms, the VMS Accounting Utility, and the VMS Monitor Utility. The following sections describe security alarms. The Accounting and Monitor Utilities are discussed in the *VMS Accounting Utility Manual* and *VMS Monitor Utility Manual*.

4.8.2.1 Security Alarms

The security manager can select one or more types of events that warrant special attention when they occur. The security manager then directs the system to send an alarm to the terminals enabled as security operator terminals when such an event is detected. For example, the security manager might identify one or more files where WRITE access should be prohibited. An alarm can be set to indicate attempted penetration to these files.

Following are events whose occurrence can trigger an alarm:

- Selected types of access to selected files and global sections
- Event requested by an ACL on a file or global section
- Use of privilege to access files and global sections
- Installation of images
- Logins, logouts, login failures, and break-in attempts
- Modifications to the system authorization file and network proxy file
- Changes to system and user passwords
- Modifications to the rights database
- Execution of the SET AUDIT command
- Volume mounts and dismounts

The security manager uses the DCL command SET AUDIT to enable security alarms. Alarms can be added or removed as necessary.

Enabling too many alarms may result in the failure to monitor each alarm appropriately. While alarms can be a powerful tool when used judiciously, they quickly lose their attention-getting quality when overused.

If you suspect your account has been broken into, change your password. You may then want to request that your security manager implement an alarm. Once an alarm has been introduced, check with your security manager periodically to see if any additional break-ins have occurred.

4.8.2.2 Auditing Access to Sensitive Files

If you feel you have key files that may have been improperly accessed, you may want to develop a strategy with your security manager to audit access to the file.

Once you have reviewed the situation and ensured that you have done everything possible to protect your files with standard UIC-based protection and a general access control list, you may conclude that a security alarm is required. To specify a security alarm, you must include a security alarm ACE in the ACL for the file. The security manager then sets up the system's security auditing to enable alarms when files are accessed.

If you suspect break-in attempts on your account, the security manager might temporarily enable an alarm for all file accesses. The security manager can also set an alarm to monitor READ access to your files to catch file browsers.

For example, if user RWOODS and his security manager concur that they must know when a highly confidential file CONFIDREVIEW.MEM is being accessed, RWOODS would add an ACE to the existing ACL for the file CONFIDREVIEW.MEM, as follows:

```
$ SET ACL CONFIDREVIEW.MEM /ACL=-
_$ (ALARM=SECURITY,ACCESS=READ+WRITE+DELETE+CONTROL+FAIL+SUCCESS)
```

The security manager would enter the following DCL commands at a terminal set up for security alarms:

```
$ REPLY/ENABLE=SECURITY
$ SET AUDIT/ALARM/ENABLE=ACL
```

The first command causes the terminal that issues the command to be enabled as a security operator. In this example, no other terminals have been established as security operator terminals. The second command enables the auditing of security alarms for file accesses involving access control list alarm ACEs. The second command could be entered by the security manager from any terminal. If user ABADGUY accesses CONFIDREVIEW.MEM with DELETE access, the following alarm appears at the security operator terminal (enabled in the previous example):

```
%%%%%%%%%%  OPCOM  30-DEC-1988 07:21:11:10  %%%%%%%%%%

Security alarm     / Successful file access
       Time:        30-DEC-1988 07:21:10.84
       PID:         23E00231
       User Name:   ABADGUY
       Image:       DUA0:[SYS0.][SYSEXE]DELETE.EXE
       File:        DUA1:[RWOODS]CONFIDREVIEW.MEM
       Mode:        DELETE
       Privs Used:  SYSPRV
```

The alarm reveals the name of the perpetrator, the method of access (successful deletion accomplished by using the program [SYSEXE]DELETE.EXE), time of access (7:21 a.m.), and the use of a privilege (SYSPRV) to gain access to the file. With this information, the security manager can take corrective action.

Note that the security alarm appears at each terminal enabled as a security operator, which in this case is only a single terminal, every time any file is accessed and meets the conditions specified in the alarm ACE for that file. CONFIDREVIEW.MEM, as well as all files on the system protected with security alarm ACEs, triggers the alarm.

An access violation of one file frequently indicates access problems with other files. Therefore, the security manager may need to monitor access to all key files having security alarm ACEs. When undesired access is being gained to key files, the security manager should take immediate action.

4.9 Managing Your Files for Optimum Security

Proper file management is an important aspect of file protection.

- Do not use obvious names for your directories and the key files in them. For example, do not title the file that contains a salary planning memo, SALARYPLAN.MEM.
- Purge your files regularly. Delete unnecessary files. This keeps your directories to a minimum and simplifies the task of regularly checking the protection and ownership on your files.
- Use the DCL command DIRECTORY/SECURITY regularly to monitor the ownership, protection code, and ACLs on your files. A user who succeeds in obtaining sufficient privilege may change the protection or ownership on your files allowing access immediately and in the future. If you perform these checks frequently, you can detect and report unexplained changes in file protection or ownership.
- Pay special attention to the protection on your mail files; normally they should be accessible only to you and the system (for mail delivery and backups).
- When you place ACLs on your files, be sure you know exactly which users hold the identifiers you have specified. (This generally requires consultation with your security manager.)
- Follow your security manager's recommendations to prevent disk scavenging. You may be requested to use the /ERASE qualifier on the SET FILE, DELETE, and PURGE commands for some or all of your files.
- Always protect files and directories that contain command procedures and executable programs. Carefully control the granting of WRITE access to these directories and files. This is particularly important if you have any of the more powerful privileges or access to sensitive files.
- Do not run a command procedure or program given to you by another user unless you inspect it. Inspect a program or procedure to see if it tries to exercise your special privileges or access sensitive files. Programs or command procedures offered under one guise, when actually intended to penetrate your defenses and disrupt your system security, are sometimes called *Trojan Horse programs*, because they parallel the theme of that Greek legend.

5 Implementing System Security

This chapter explains how security managers can implement security features of the VMS operating system. Descriptions are based on the security needs of a commercial installation with average security needs, such as files and accounts requiring protection. Descriptions of above-average security needs are noted as such.

DIGITAL recommends that you read the entire chapter before establishing any security measures. After reading the chapter, you will better be able to decide which security measures are appropriate for your site, and you will have the tools to implement them.

The Authorize Utility (AUTHORIZE) is the primary tool for implementing system security. AUTHORIZE is described fully in the *VMS Authorize Utility Manual*. The System Generation Utility (SYSGEN) and many DCL commands are also important security tools. For more information about SYSGEN, see the *Guide to Setting Up a VMS System* and the *VMS System Generation Utility Manual*. DCL commands are described in the *VMS DCL Dictionary*.

5.1 Security Management Account

Security managers require accounts with privileges. In many cases, the security manager and system manager roles are performed by the same person, so the same account can serve both. The *Guide to Setting Up a VMS System* describes the necessary characteristics of a system management account. It is important that strong cooperation and open communication exist when the security management and system management roles are performed by separate individuals.

The security manager requires the additional SECURITY privilege to enable security auditing and to set up security operator terminals.

5.2 Considerations for Establishing User Accounts

The security manager performs user training, assigns accounts and passwords, and monitors user actions. This section describes guidelines for all interaction with users.

A security manager bases decisions about setting up user accounts on the needs of the particular site. You have the option to develop one or more templates that work for many of your users. However, do not oversimplify the process of account creation to the point that you simply apply a template. The danger in relying solely on templates is that you might overlook special considerations that apply to individual users, thereby forfeiting important controls that only you can exercise.

Examine templates regularly to be sure they are valid and reflect the way you want your operations to proceed. Templates become obsolete rapidly.

5.2.1 Introduction to Group Design

As you design user groups, remember that the groups you establish have an impact on file protection and influence those who receive the GROUP, GRPNAM, and GRPPRV privileges. First, you may want to map out the functions you expect your users to perform. Look for groups of users involved with a common function, such as accounting, engineering, marketing, and personnel.

Think ahead to future plans in your organization. Incorporate these ideas into your strategy. You can fine tune the group design at any time, but it is most important to gain a perspective on the logical groupings according to the functions your users perform.

The following example illustrates many principles of group design. The Rainbow Paint Company is a distribution company with five departments: executive, accounting, marketing, shipping, and secretarial. Table 5–1 identifies the employees in those departments who need computer resources, and their job responsibilities.

Table 5–1 Employee Grouping by Department and Function

Department	Employee	Function
EXECUTIVE	Sunny Gold	President
	Olive Green	Treasurer Head of Computer Operations
ACCOUNTING	Lily White	Payroll
	Rich Silver	Bookkeeping
	Red Orange	Clerk
	Ruby Crimson	Clerk
MARKETING	Rusty Brown	Forecasting
	Tawny Copper	Sales Reporting
SHIPPING	Sky Blue	Inventory Control
SECRETARIAL	Misty Grey	Correspondence Management Paycheck Printing

The fact that the company has been organized into departments suggests that individuals in the same department perform many of the same functions. For example, the advantage of grouping all the employees who "keep the books" for the company in the accounting department is that employees can easily gain access to one another and to the data they must share.

As the system manager of Rainbow Paint's computer resources, Olive Green will set up UIC groups based on the existing organizational structure. For example, the employees in the accounting department (L. White, R. Silver, R. Orange, and R. Crimson) could be members of the UIC group ACCOUNTING. Setting up the UIC group in this way ensures that user L. White has easy access to data from user R. Silver, and so forth.

Effective department organization ensures that only selected employees will have access to all data and employees in the company. For example, one of the functions of the accounting department concerns payroll. Because payroll information is confidential, employees in the shipping and marketing departments should not have access to that information.

Following are two guidelines for determining the placement of users in UIC groups:

- Sharing—Users who typically share data and control of processes should be arranged in the same group.
- Protection—Users who should not have access to each other's data or control each other's processes should be assigned to separate groups.

As the system/security manager of Rainbow Paint's computer resources, Olive sets up the UIC groups ACCOUNTING, EXECUTIVE, MARKETING, SHIPPING, and SECRETARIAL corresponding to the various departments in the company. Members of a UIC group can be given common access to files, as shown in the following example:

```
$ SET PROTECTION=G:RWE GROUP_STATS.DAT
```

In this command, the owner of the file GROUP_STATS.DAT allows each member of the UIC group READ, WRITE, and EXECUTE access to the file.

However, there are limitations to UIC group design. You may want to give only a few members of your UIC group access to files that you own, or you may want to grant access to your files to members of several UIC groups without having to grant world access. These limitations are described in the next section.

5.2.1.1 Limitations to UIC Group Design

In some cases, UIC-based protection does not present the best solution to your file protection needs. If users in several UIC groups need access to common files on the system, the only UIC-based alternatives are to give world access to the object (all users can access the object) or to grant extended privileges to each user. Neither choice is desirable.

You may also need to allow users in a UIC group several types of access to files; you may want to deny some users in the same group access to the object. Again, UIC-based protection does not offer a good solution to meet these needs.

Access control lists (ACLs), described in the following sections, offer an alternative means of protecting files and other objects on the system.

5.2.2 Introduction to ACL Design and Identifiers

Rather than attempting to restructure UIC groups to solve file protection problems, you may be able to achieve your goals by using access control lists on the files. For example, consider the ACL that you might construct to allow specific users (across various UIC groups) to access the file PAYROLL.DAT:

```
(IDENTIFIER=OGREEN,ACCESS=READ+WRITE+EXECUTE+DELETE)
(IDENTIFIER=LWHITE,ACCESS=READ+WRITE+EXECUTE+DELETE)
(IDENTIFIER=RSILVER,ACCESS=READ+WRITE+EXECUTE+DELETE)
(IDENTIFIER=MGREY,ACCESS=READ)
(IDENTIFIER=SGOLD,ACCESS=READ)
```

Notice that many of the users share the same access needs. To shorten the ACL, you could use AUTHORIZE to define a general identifier PAYROLL in the rights database. The holders of that identifier could be all users who need RWED access to PAYROLL.DAT. Once the identifier and its holders are

defined, you could use the following ACL to specify the same type of access to PAYROLL.DAT:

```
(IDENTIFIER=PAYROLL,ACCESS=READ+WRITE+EXECUTE+DELETE)
(IDENTIFIER=MGREY,ACCESS=READ)
(IDENTIFIER=SGOLD,ACCESS=READ)
```

The VMS operating system processes shorter ACLs more rapidly. Also, when employees change but the functions remain the same, you do not have to change every ACL across the system. Instead, remove the user's UAF record; as a consequence, that user also no longer holds the identifier. When a new employee is hired for the same job, grant the new user the right to hold the identifier. The new user then has the same ACL-based access as the former user.

What you want most of all is an overall design that considers the types of files on your system and the protection needs of each. If you have successfully designated groups and identifiers, you should be able to easily design ACLs and define standard protection. Time spent clarifying the common access needs of your users simplifies the design of identifiers and ACLs. You will also simplify the job for your users who place ACLs on their files.

Do not use ACLs indiscriminately. They can consume large amounts of paged system dynamic memory when files are open. They may also require additional processing time. ACLs are best applied where protection is really needed.

For more information on defining identifiers, see the description of the VMS Authorize Utility in the *VMS Authorize Utility Manual*. For more information about creating and maintaining ACLs, see the VMS ACL Editor description in the *VMS Access Control List Editor Manual*.

5.2.3 Some Special-Purpose Identifiers

The system provides a group of reserved identifiers: LOCAL, DIALUP, REMOTE, INTERACTIVE, NETWORK, and BATCH. These identifiers permit you to define a large potential group of users according to their use of the system. Typically, these types of identifiers are used in combination with other identifiers. For example, the following ACE permits user Martin to have RWED access to the object only when logged in from a local terminal:

```
(IDENTIFIER=MARTIN+LOCAL,ACCESS=READ+WRITE+EXECUTE+DELETE)
```

You can use the reserved system identifiers in ACLs to deny access to an entire class of logins. For example, the following ACE denies access to all dialup users:

```
(IDENTIFIER=DIALUP,ACCESS=NONE)
```

5.2.4 Creating and Maintaining a Rights Database

Once you have designed the names of the identifiers you want on your system and composed the set of holders for the identifiers, you must use the VMS Authorize Utility to make these associations known to the system. These associations are kept in the rights database (RIGHTLIST.DAT), which you maintain as you add or remove users and identifiers.

The rights database is initially created for every VMS system at system installation and is located in the SYS$SYSTEM directory. At that time it contains the names of the reserved system identifiers and one identifier for each authorized user. The identifier, called a UIC identifier, is associated with the user's UIC and user name.

There is also an identifier in the rights database equivalent to each UIC group name. When you add a new user as the first member of a new UIC group, and you specify an account name with the user, an identifier corresponding to the account name is added to the rights database, as shown in the following example:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD ROB/PASSWORD=SPO152/UIC=[014,006] -
_UAF> /DIRECTORY=WORK:[ROB]/ACCOUNT=MGMT
UAF-I-ADDMSG, user record successfully added
UAF-I-RDBADDMSGU, identifier ROB value: [000014,000006] added to RIGHTSLIST.DAT
UAF-I-RDBADDMSGU, identifier MGMT value: [000014,177777] added to RIGHTSLIST.DAT
```

Each site adapts its own rights database according to actual use and needs.

Note that when you use AUTHORIZE to add, remove, or change user names in the system UAF, AUTHORIZE makes corresponding changes for you in RIGHTSLIST.DAT, so that the rights database corresponds to the system UAF.

You will seldom need to use the AUTHORIZE command CREATE/RIGHTS because of the automatic creation and maintenance of the rights database. However, if the rights database is damaged or deleted, you can create a new one with this command.

5.2.4.1 Adding Identifiers

Add identifiers with the AUTHORIZE command ADD/IDENTIFIER, as follows:

```
UAF> ADD/IDENTIFIER PAYROLL
identifier PAYROLL value %X80080011 added to RIGHTSLIST.DAT
```

If you accidentally deleted the rights database, and it cannot be recovered from a backup copy, recreate RIGHTSLIST.DAT by entering the CREATE/RIGHTS command and then an ADD/IDENTIFIER command, as follows:

```
UAF> CREATE/RIGHTS
{message}
UAF> ADD/IDENTIFIER/USER=*   or   ADD/IDENTIFIER/USER=[*,*]
{messages}
```

The ADD/IDENTIFIER command generates a UIC identifier in the rights database corresponding to each user name in the UAF. To complete the task, use the ADD/IDENTIFIER command to add all general identifiers that were lost. Then redefine the holders of the identifiers with GRANT/IDENTIFIER commands, as shown in the next section.

5.2.4.2 Adding Holders of Identifiers

Associate users as holders of existing identifiers using the AUTHORIZE command GRANT/IDENTIFIER, as shown in the following example:

```
UAF> GRANT/IDENTIFIER PAYROLL MARTIN
UAF-I-GRANTMSG, identifier PAYROLL granted to MARTIN
UAF> GRANT/IDENTIFIER PAYROLL STEVENS
UAF-I-GRANTMSG, identifier PAYROLL granted to STEVENS
```

To give MARTIN the EXECUTIVE identifier would require another use of the GRANT/IDENTIFIER command. You can only introduce one holder association at a time with the GRANT/IDENTIFIER command.

5.2.4.3 Removing Identifiers and Holders

When a user leaves the company, remove the UAF record for that user. Notify the managers of all sites where that user has access to proxy accounts to remove proxy access information in the remotes node's NETPROXY.DAT file. When you run AUTHORIZE to remove a user's UAF record, AUTHORIZE also removes the user's connections as a holder of identifiers in the rights database. However, if a departed user is the only remaining holder of a given identifier, remove that identifier to avoid future confusion.

For example, use the following AUTHORIZE command to remove the identifier 87TERM3:

```
UAF> REMOVE/IDENTIFIER 87TERM3
{message}
```

Before you remove an identifier from the rights database, remove all occurrences of the identifier from ACLs on the system. There is no single command that displays all uses of an identifier in the ACLs.

The following example explains how to remove the obsolete identifier 87SUMMER from the ACL on file TESTPROG.EXE. Invoke the VMS ACL Editor to display the contents of the ACL:

```
$ EDIT/ACL TESTPROG.EXE
(IDENTIFIER=STAFF,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=87SUMMER,ACCESS=READ+WRITE+EXECUTE)
(IDENTIFIER=)
```

The second ACE in the ACL contains the identifier 87SUMMER. Position the cursor at the second ACE and press the PF4 key on the keypad to delete the line. Press CTRL/Z to exit the VMS ACL Editor. The successfully updated ACL is shown, as follows:

```
$ EDIT/ACL TESTPROG.EXE
(IDENTIFIER=STAFF,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=)
CTRL/Z
%ACLEDIT-S-ACLUPDATED, ACL successfully updated
$
```

Repeat this procedure for all ACLs containing the identifier 87SUMMER. Next, remove the identifier 87SUMMER from the rights database with the AUTHORIZE command REMOVE/IDENTIFIER.

Identifiers in hexadecimal format in an ACE indicate that a general identifier has been deleted from the rights database. Similarly, if you see an identifier displayed as a numeric UIC, the original identifier was a UIC that has been removed. Delete ACEs with numeric UIC or hexadecimal identifiers.

Do not reuse UICs too soon after an employee leaves. The new employee may gain some or all of the access rights of the previous employee through ACL entries that still reference the old UIC in numeric format.

To rename an identifier, use the AUTHORIZE command RENAME/IDENTIFIER in the following format:

```
UAF> RENAME/IDENTIFIER old-identifier new-identifier
```

Renaming identifiers affects existing ACLs; those that have ACEs specifying renamed identifiers are updated with the new identifier name.

5.2.4.4 Displaying the Rights Database

Security managers should regularly display the rights database to check that it is correct and current. Two AUTHORIZE commands are used for this: SHOW/IDENTIFIER and SHOW/RIGHTS. To display all holders of an identifier, use the SHOW/IDENTIFIER command, as shown in the following example:

```
UAF> SHOW/IDENTIFIER/FULL identifier-name
```

Use the wildcard asterisk character to display all holders of all identifiers on the system, as follows:

```
UAF> SHOW/IDENTIFIER/FULL *
```

To display the identifiers held by a particular user, use the SHOW/RIGHTS command, as shown:

```
UAF> SHOW/RIGHTS/USER=username
```

Use the wildcard asterisk character to display all identifiers held by all users, as follows:

```
UAF> SHOW/RIGHTS/USER=*
UAF> SHOW/RIGHTS/USER=[*,*]
```

The first command displays users alphabetically. The second command displays users according to UICs.

5.2.5 Setting Protection and Ownership Defaults for Users

If you know that a user will be using a directory or file that demands special protection, you must determine which of a number of possible defaults will affect the user. Exert additional control where the default protection is deemed inadequate. Section 4.5 describes default protection in detail. There are three possible areas where security managers can specify protection defaults that would affect the user. In the order of increasing influence, they are as follows:

- The systemwide default for file protection provided by the SYSGEN parameter RMS_FILEPROT. This value always exists and may be the only one to influence the outcome. Security managers can change the value of RMS_FILEPROT with SYSGEN. However, the effectiveness of this value may be overridden by either of the following.
- The DCL command SET PROTECTION/DEFAULT (that can be placed in the user's login command procedure) can specify the file protection placed on files created or modified by the user during the terminal session. At any time during a session, the user can also enter this command to override the value set by a previous SET PROTECTION/DEFAULT

command. The SET PROTECTION/DEFAULT command negates the influence of the systemwide protection for this user.

- The default protection for the specific directory can be specified in an ACL applied to the directory. If the DEFAULT_PROTECTION ACE exists for the directory, all new files added to the directory, including subdirectories and their files, are subject to this protection code. This code overrides the systemwide default and the user-specified default (if any).

You may want to include the protection imposed on the volume through the DCL command SET VOLUME/PROTECTION. This protection code, if specified, prevents a user from accessing any part of the volume, regardless of the protection code on the directory or the file. If no volume protection is specified with the SET VOLUME command, the volume is open to all users.

As Section 4.4 explains, the assignment of file ownership affects the outcome of any protection check. The operational effect of this combined protection structure is depicted in Figure 5–1.

Figure 5–1 Flowchart of File Creation

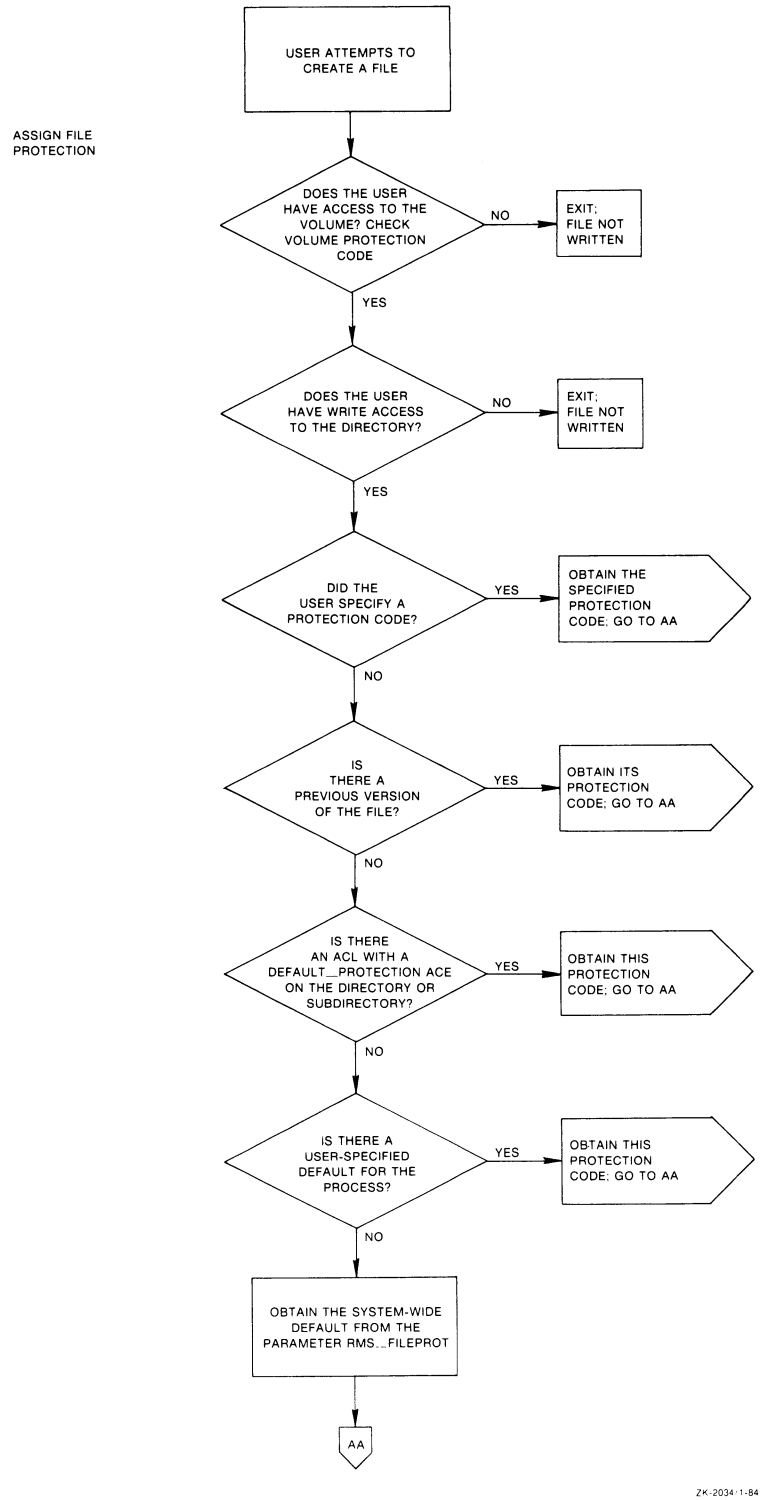


Figure 5–1 Cont'd. on next page

Figure 5–1 (Cont.) Flowchart of File Creation

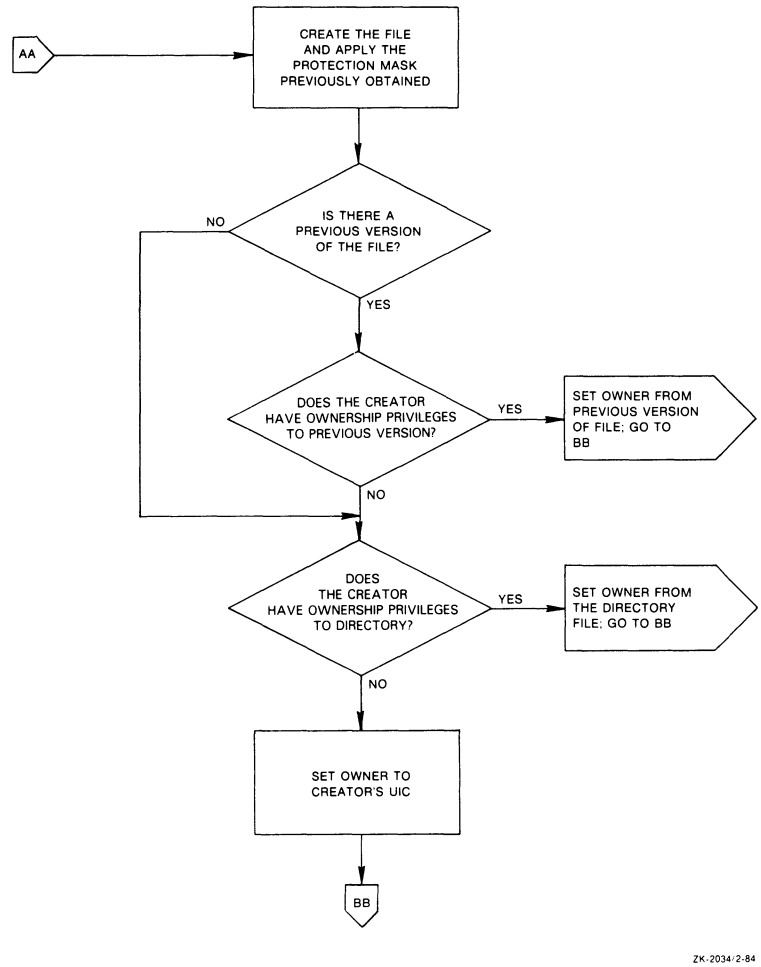
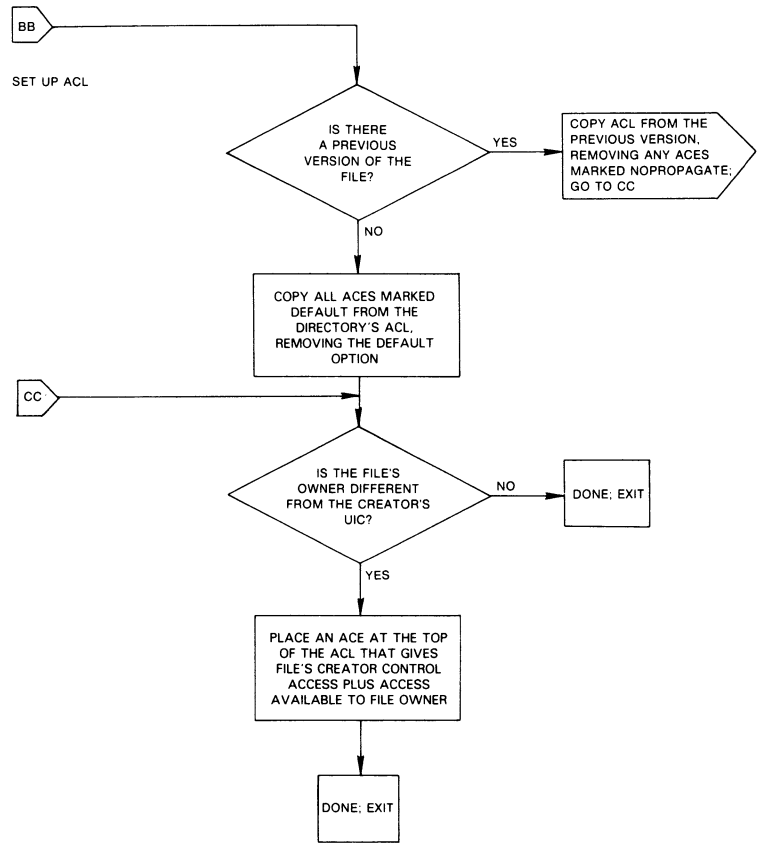


Figure 5–1 Cont'd. on next page

Figure 5–1 (Cont.) Flowchart of File Creation



5.2.5.1 Adjusting Protection Defaults

You may want to make adjustments to control default behavior. The systemwide default protection code specified by the SYSGEN parameter RMS_FILEPROT sets the user's default protection to:

(S:RWED,O:RWED,G:RE,W)

Assume that the volume protection has been set by the operator to:

(S:RWED,O:RWED,G:R,W)

The file protection on the directory [PROJECT] has been set to:

(S:RWED,O:RW,G:R,W)

If all the files created in the subdirectory [PROJECT.DIARY] demand more protection, you, or any user who has the same access as the owner of the directory, could define a specific default protection code for this specific directory with an ACL consisting of a DEFAULT_PROTECTION ACE, as follows:

(DEFAULT_PROTECTION,S:RWED,O:RWED,G,W)

The following DCL command would provide the desired default protection:

```
$ SET ACL/ACL=(DEFAULT_PROTECTION,S:RWED,O:RWED) [PROJECT]DIARY.DIR
```

Once this ACL is placed on the directory file, all files created or modified in the directory will be subject to the default protection code. However, default protection codes can be easily overridden by any user with the BYPASS privilege and can be circumvented under certain circumstances by users with SYSPRV, GRPPRV, or READALL privileges. Because they are only defaults, a user who acquires CONTROL access to a file in the directory can include a specific protection code as a replacement for the default value on the file using the following DCL commands:

- SET PROTECTION
- COPY/PROTECTION
- APPEND/PROTECTION

Once the default protection code is replaced, the new code becomes the default and is propagated to subsequent versions of the file.

If you provide a special login command procedure for some of your users, you may want to supplement the systemwide default process protection specified by the SYSGEN parameter RMS_FILEPROT for this group of users. Add the SET PROTECTION/DEFAULT to the login command procedure to specify the default process protection, as follows:

```
SET PROTECTION=(S:RWED,O:RWED,G,W)/DEFAULT
```

Files created in the users' directories receive this default protection code unless explicitly overridden.

5.2.5.2 Setting Up a Project Account

To allow for more flexible management and accounting of disk space, identifiers can carry the optional resource attribute. This attribute, when present on an identifier, allows file space to be owned by and charged to that identifier. Thus, when a project or department-specific identifier is the owner of a directory, the space used by files created in the directory can be charged to the appropriate department or project rather than to the individual who created them. When users work on multiple projects, they can charge their disk space requirements to the related project rather than to their personal accounts.

Example

To set up a project identifier and directory, perform the following steps:

1. Create the project identifier with the resource attribute in the rights database. The following example creates the identifier PROJECTX:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> ADD/IDENTIFIER PROJECTX /ATTRIBUTES=RESOURCE
```

2. Grant the identifier to the appropriate individuals with the resource attribute.

```
UAF> GRANT/IDENTIFIER PROJECTX user1 /ATTRIBUTES=RESOURCE
UAF> GRANT/IDENTIFIER PROJECTX user2 /ATTRIBUTES=RESOURCE
   .
   .
   .
```

3. Create the disk quota authorization for the project identifier. For example, the following command invokes the VMS System Management (SYSMAN) Utility and assigns the identifier PROJECTX 2000 blocks of disk quota with 200 blocks of overdraft:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> DISKQUOTA ADD PROJECTX /PERMQUOTA=2000 /OVERDRAFT=200
```

4. Create the project directory. For example, the following DCL command creates the project directory [PROJECTX] and establishes the identifier PROJECTX as the owner:

```
$ CREATE/DIRECTORY [PROJECTX] /OWNER=[PROJECTX]
```

5. Set up the necessary ACL on the project directory to allow holders of the PROJECTX identifier access to the directory. For example, the following DCL command places an ACL on the directory [PROJECTX] that permits any holder of the identifier PROJECTX to gain READ, WRITE, or EXECUTE access to the directory. The second ACE specifies that files created in the directory will receive the same ACE as a default.

```
$ SET DIRECTORY [PROJECTX] /ACL= (-
_$ (IDENTIFIER=PROJECTX,ACCESS=READ+EXECUTE),-
_$ (IDENTIFIER=PROJECTX,OPTIONS=DEFAULT,ACCESS=READ+EXECUTE))
```

Access must be granted through ACL entries, since the owner identifier of the directory and the files does not match the UIC of any of the project members; thus, only SYSTEM and WORLD access are available through the UIC-based protection mask. The first ACE of the specified ACL gives all project members READ and EXECUTE access to the directory; the second ACE gives the same access for all files created in the directory. (The DEFAULT option in the

second ACE specifies that the ACE is to be copied to each file created in the directory.)

Note that project members are not allowed to delete (or control) files created by others. However, the members each have complete access to files they have created in the directory, because the file system supplies an additional ACE that grants the file creator CONTROL access plus the access specified in the OWNER field of the UIC-based protection mask. (The flowchart in Figure 5–1 illustrates how the OWNER field is derived.) This ACE only appears when the owner of the created file does not match the UIC of the creator, as is the case for files created in an account owned by a project identifier.

Thus, when project member CRANDALL creates files in the [PROJECTX] directory, the files receive the following access control list:

```
(IDENTIFIER=CRANDALL,OPTIONS=NOPROPAGATE,ACCESS=READ+WRITE+EXECUTE+DEFAULT+CONTROL)
(IDENTIFIER=PROJECTX,ACCESS=READ+EXECUTE)
```

This example assumes that the OWNER field grants full (RWED) access. Because this may not always be true (the systemwide default set by the SYSGEN parameter RMS_FILEPROT may have been changed, or a user may have specified a process-specific default protection mask with the DCL command SET PROTECTION/DEFAULT), you may want to ensure consistency by providing a default protection ACE in the project directory ACL, as follows:

```
$ SET DIRECTORY [PROJECTX] /ACL= (-
_$ (DEFAULT_PROTECTION,S:RWED,O:RWED,G,W),-
_$ (IDENTIFIER=PROJECTX,ACCESS=READ+EXECUTE),-
_$ (IDENTIFIER=PROJECTX,OPTIONS=DEFAULT,ACCESS=READ+EXECUTE))
```

The UIC protection specified in the default protection ACE is applied to all files created in the project directory.

5.2.6 Password Management

A site needing average security protection always requires use of passwords. Sites with more security needs frequently impose a double password scheme (see Section 5.2.6.3) requiring primary and secondary passwords, and possibly system passwords as well.

This section describes password management.

5.2.6.1 Initial Passwords

When you open an account for a new user with AUTHORIZE, you must give the user a user name and an initial password. When you assign temporary initial passwords, observe all guidelines recommended in Section 3.1.3.10. You may want to use the automatic password generator. Aviod any obvious pattern when assigning passwords.

To use the automatic password generator while using the VMS Authorize Utility to open an account, add the /GENERATE_PASSWORD qualifier to either the ADD or COPY command. The system responds by offering you a list of automatically generated password choices. Select one of these passwords, and continue setting up the account.

When you add a new user to the UAF, you may want to define that user's password as having expired previously using the AUTHORIZE qualifier /PWDEXPIRED. This forces the user to change the initial password when first logging in. The system behaves just as if the password had reached its expiration date, as described in Section 5.2.6.4.

Preexpired passwords are conspicuous in the UAF record listing. The entry for the date of the last password change carries the following notation:

<pre-expired>

By default, VMS forces new users to change their password the first time they log in. Include the first login as part of your user training.

5.2.6.2 System Passwords

Section 3.1.3.1 introduces system passwords, which control access to particular terminals. System passwords are used to control access to terminals that might be targets for unauthorized use, as follows:

- All terminals using dialup lines or public data networks for access
- Terminals on lines that are publicly accessible and not tightly secured, such as those at computer laboratories at universities
- Terminals not frequently inspected
- Terminals intended for use only as spare devices
- Terminals the security manager wants to reserve for security operations

Implementing system passwords is a two-stage operation involving the DCL commands SET TERMINAL and SET PASSWORD. First, you must decide which terminals require system passwords. Then, for each terminal, you enter the DCL command SET TERMINAL/SYSPWD/PERMANENT. When you are satisified that you have selected the right terminals, incorporate these commands in SYS$MANAGER:SYSTARTUP.COM so that the terminal setup work is done automatically at system startup time. You can remove the restriction on a terminal at any time by invoking the DCL command SET TERMINAL/NOSYSPWD/PERMANENT for that terminal.

Then choose a system password and implement it with the DCL command SET PASSWORD/SYSTEM, which requires the SECURITY privilege. This command will prompt you for the password and then ask you to reenter it for verification, just as is done for user passwords. To request automatic password generation, include the /GENERATE qualifier.

To enable the use of the system password for the remote class of logins (those accomplished through the DCL command SET HOST), set the appropriate bit in the default terminal characteristics parameter using SYSGEN. This is bit 19 (hexadecimal value 80000) in the parameter TTY_DEFCHAR2. Note that if you set this bit, you must invoke the DCL command SET TERMINAL/NOSYSPWD/PERMANENT to disable system passwords for each terminal where you do not want the feature. (As before, consider placing the SET TERMINAL commands you have tested in SYS$MANAGER:SYSTARTUP.COM.) Follow the steps in the preceding paragraph to set the system password.

When choosing a system password, follow the recommendations of Section 3.1.3.10. Choose a non-English string of characters and digits, with a minimum length of 6. The system password is not subject to expiration. Change the password frequently. Always change the system password as soon as a person who knows the password leaves. Share the system password only with those who need to know.

The system password is stored in a separate UAF record and cannot be displayed. The DCL command SET PASSWORD/SYSTEM (the normal means of setting and changing the system password) requires that you enter the old system password prior to changing it. Use the AUTHORIZE command MODIFY/SYSTEM_PASSWORD to change the system password without having to specify the old password, as shown in the following command:

```
UAF> MODIFY/SYSTEM_PASSWORD=ABRACADABRA
```

The primary function of the system password is to form a first line of defense for publicly accessible ports and to prevent potential intruders from learning the identity of the system. However, requiring system passwords can appear unfriendly when authorized users are unaware that they are required on certain terminals. To avoid false reports of defective terminals or systems, inform your users which terminals allocated for their use require system passwords.

Where system passwords are not applied to either control access through dialup lines or on publicly accessed lines, few people might know the system password. There is the possibility of encumbered operations if the personnel who know the password are unavailable, incapacitated, or forget the password. Solve this problem by invoking AUTHORIZE and entering the MODIFY/SYSTEM_PASSWORD command. SYSPRV privilege is required.

5.2.6.3 Primary and Secondary Passwords

The use of dual passwords is cumbersome and mainly warranted at sites with high-level security concerns. Dual passwords offer three advantages: when used on a widespread basis, they facilitate the verification of the physical identity of each user at login time through visual contact; when used in limited cases, they single out accounts that can only be logged in to when two persons are present; they also prevent accounts from being accessed through DECnet using simple access control.

Sites with medium security requirements might want to use dual passwords as a tool when there are unexplained break-ins after the password has been changed and the use of the password generator has been enforced. Select problem accounts, and make them a temporary target of this restriction. If the problem goes away when you institute personal verification through the secondary password, you know you have a personnel problem. Most likely, the authorized user is revealing the password for the account to one or more other users who are abusing the account.

Implement dual passwords with the AUTHORIZE qualifier /PASSWORD. For example, to impose dual passwords on a new account, invoke AUTHORIZE and enter the following command:

```
UAF> ADD newusername /PASSWORD=(primarypwd, secondarypwd)
```

To impose a secondary password on an existing account, enter the following command:

```
UAF> MODIFY username /PASSWORD=("", secondarypwd)
```

This command does not affect the primary password that already exists for the account, but adds the requirement that a secondary password be provided at each subsequent login. The secondary password acquires the same password lifetime and minimum length values in effect for the primary password. If the /FLAGS=GENPWD qualifier has been specified for this account, the secondary password can be changed only under the control of the automatic password generator.

Note: While secondary passwords can be specified for accounts requiring remote access using the DCL command SET HOST, they cannot be specified for accounts requiring network file access using access control strings. Do not specify secondary passwords on accounts that require network access, or request remote security managers to set up proxy accounts for those users requiring file access to other nodes in the network.

5.2.6.4 Enforcing Minimum Password Standards

Security managers can use AUTHORIZE to impose minimum password standards for individual users. Specifically, qualifiers and login flags provided by AUTHORIZE control the minimum password length, how soon passwords will expire, and whether the user is forced to change passwords at expiration.

Password Expiration

With the AUTHORIZE qualifier /PWDLIFETIME, you can establish the maximum length of time that can elapse between password changes before the user will be forced to change the password or lose access to the account. By default, the value of /PWDLIFETIME is 180 days. You can change the frequency requirements for user password changes by specifying a different delta time value for the qualifier. For example, to require a user to change the password every 60 days, you would specify the qualifier as /PWDLIFETIME=60-0.

The /PWDLIFETIME qualifier applies to both primary and secondary user passwords, but not to the system password. Each primary and secondary password for a user is subject to the same maximum lifetime. However, the passwords can change at separate times. As soon as the user completes a password change, that individual password's clock is reset; the new password value can exist unchanged for the length of time dictated by /PWDLIFETIME.

The Authorize Utility also provides two login flags related to primary and secondary password expiration. These flags, PWD_EXPIRED and PWD2_EXPIRED, are specified with the /FLAGS qualifier. The first flag, PWD_EXPIRED, is set on after the primary password expires and the user has had one last chance to change the password and failed to do so. The second flag, PWD2_EXPIRED, is set on after the secondary password expires, and the user has had one last chance to change the secondary password and failed to do so. If either PWD_EXPIRED or PWD2_EXPIRED is set, the account will be disabled for logins, since the user failed to employ the last chance to change the password during the last login.

As soon as the user successfully changes the password, VMS resets the flags, as appropriate. The flag PWD_EXPIRED becomes NOPWD_EXPIRED as soon as the primary password is changed. Similarly, the flag PWD2_EXPIRED becomes NOPWD2_EXPIRED as soon as the secondary password is changed. As security manager, you may choose to invoke AUTHORIZE and reset the flags, giving the user another chance to reset the password.

The use of a password lifetime forces the user to change the password regularly. The lifetime can be different for different users. Users with access to critical files generally should have the shortest password lifetimes.

System passwords have an unlimited lifetime. Therefore, change the system password regularly.

Forcing Expired Password Changes

By default, users are forced to change expired passwords when logging in. Users whose passwords have expired are prompted for new passwords at login. This password feature is only valid when a password expiration date is specified with the /PWDLIFETIME qualifier.

To disable forced password changes, specify the following qualifier to the ADD or MODIFY command:

```
/FLAGS=DISFORCE_PWD_CHANGE
```

Once disabled, the forced password feature can be reenabled by clearing the login flag, as shown in the following example:

```
/FLAGS=NODISFORCE_PWD_CHANGE
```

Users who log in and are prompted to change expired passwords can abort the login by pressing CTRL/Y.

Note: If secondary passwords are in effect and both primary and secondary passwords have expired, the user is forced to change both passwords. If the user changes the primary password and presses CTRL/Y before changing the secondary password, the user is logged out, and no password change is recorded.

Minimum Password Length

With the AUTHORIZE qualifier /PWDMINIMUM, you can direct that all password choices must be a minimum number of characters in length. Users can still specify passwords up to the maximum length of 31 characters.

This length applies both to primary and secondary passwords and is only required when users change passwords with the DCL command SET PASSWORD. As security manager, you can specify initial passwords through AUTHORIZE that are shorter than the minimum. However, doing so may confuse your users unnecessarily. Furthermore, initial passwords inherently introduce security weaknesses. By selecting short initial passwords, you compound the problem. Generally, it is good practice to observe the same rules you expect your users to follow.

There is always a minimum password length in effect for each user. It is either the default of 6 or another value established by the /PWDMINIMUM qualifier. Thus, if the user specifies the DCL command SET PASSWORD/GENERATE=n to automatically generate new password choices, *n* must be a value at least as great as the minimum value in effect. If *n* is less than the current minimum enforced in the UAF, it is disregarded; no message appears. The five password choices that VMS generates for the user comply with the current minimum password length.

The password generator creates passwords that range in length between *n* and *n*+2, where *n* is the specified or minimum length. In addition, the maximum values for *n* and *n*+2 that the password generator can accommodate are 10 and 12, respectively. Longer passwords require an inordinate amount of CPU time to generate.

The system password is not subject to a minimum length. Guidelines that apply to user passwords are equally applicable to system passwords. Choose system passwords that are 6 to 10 characters long.

5.2.6.5 Requiring the Password Generator

The /FLAGS=GENPWD qualifier in AUTHORIZE allows you to force use of the automatic password generator when a user changes a password. At some sites, all accounts will be created with this qualifier. At other sites, the security manager may be more selective.

Criteria for requiring use of the password generator should be whether or not the user will have access to sensitive data that must not be compromised by a break-in.

If your policy is to request voluntary use of the password generator, and users are not cooperating, you can force users to use the password generator by adding the /FLAGS=GENPWD qualifier to most or all user accounts. You can also add the AUTHORIZE qualifier /FLAGS=LOCKPWD to user accounts to prevent users from changing passwords. Only you as system manager will be authorized to change passwords.

5.2.6.6 Protecting Passwords

In addition to all the recommendations included in Section 3.1.3.10, the security manager should observe the following guidelines to protect passwords:

- Make certain the passwords on the standard accounts SYSTEM, FIELD, and SYSTEST are secure and changed regularly, or disable the accounts with the AUTHORIZE qualifier /FLAGS=DISUSER when they are not in use.
- Do not permit an outside or in-house service organization to dictate the password for an account they use to service your system. Such service groups tend to use the same password on all systems, and their accounts are usually privileged. On seldom-used accounts, set the AUTHORIZE flag DISUSER, and only enable the account when it is needed. You can also change the password immediately after each use, and notify the service group of the new password when they need it next.
- Delete accounts no longer in use.
- If you have an account on a system that stores passwords in plaintext (unencrypted), choose a different password on all your other accounts.
- Do not leave listings where they could be read or stolen.
- Maintain adequate protection of authorization files. Note that the system user authorization file (SYSUAF.DAT) and network proxy authorization file (NETPROXY.DAT) are owned by the system account ([SYSTEM]). There should be no other users in this group. Accordingly, the categories SYSTEM, OWNER, and GROUP are synonymous. Normally the default UIC-based file protection for these authorization files is adequate. (You might use ACLs on the listing files SYSUAF.LIS and NETPROXY.LIS to grant access only to selected individuals.)

Following are actions not strictly for password protection, but that reduce the potential of password detection or limit the extent of the damage if passwords are discovered or bypassed:

- Avoid giving multiple users access to the same account.

- Protect telephone numbers for dialup lines connected to your system.
- If your system has accounts available to outside users, such as guest accounts or Quality Assurance Reporting accounts, make these accounts captive accounts (restricted) contained by captive command procedures. (See Section 5.8 for information about setting up captive accounts.)
- Make all accounts that do not require a password captive accounts.
- Extend privileges to users carefully.
- Protect your own files using all the techniques recommended in Section 4.9.
- Ensure that the files containing components of the operating system are adequately protected (see Appendix C).
- Use the AUTHORIZE qualifiers /NOINTERACTIVE and /NOBATCH when setting up proxy login accounts to only permit file access from other nodes. Interactive and batch logins are disabled for the account.

5.2.7 Login Options

This section describes how you can control the display of various pieces of information that appear by default at login time, such as announcement, welcome, last login, and new mail messages. In addition, this section describes the use of the secure server and how to set up break-in detection and evasion.

5.2.7.1 Controlling the Announcement Message

Define the system logical name SYS$ANNOUNCE in your site-specific startup command procedure to provide an announcement message on your system. The *Guide to Setting Up a VMS System* describes how to do this. The announcement message appears at login.

The definition you provide here affects all users on the system. Because this message may provide a clue to the identity of the operating system, you may decide not to display it.

5.2.7.2 Controlling the Welcome Message

Similar to the announcement message, the welcome message is controlled through a system logical name, SYS$WELCOME. If you do not define SYS$WELCOME, a standard welcome message is provided for all users. This welcome message reveals the operating system and version number, as well as the node, if SYS$NODE is defined.

If you prefer to write your own message, you can define another message for SYS$WELCOME. To disable the welcome message, place the following DCL command in SYS$MANAGER:SYSTARTUP.COM. This command prints a blank line in place of the standard welcome message.

```
$ DEFINE/SYSTEM SYS$WELCOME "    "
```

(See the *Guide to Setting Up a VMS System* for details.)

If you prefer to selectively disable the message for individual users, you can use the AUTHORIZE qualifier /FLAGS=DISWELCOME on individual UAF records.

5.2.7.3 Controlling the Last Login Messages

You can selectively disable the appearance of the series of three messages that detail information regarding last logins and number of failed attempts at logging in. Enter the /FLAGS=DISREPORT qualifier provided with AUTHORIZE for specific users.

5.2.7.4 Controlling New Mail Announcements

You can prevent users from receiving notification that they received new mail since the last time they were logged in by specifying the AUTHORIZE qualifier /FLAGS=DISNEWMAIL. Because this has no impact on security, it is primarily a user convenience. If a user cannot invoke the VMS Mail Utility (usually applies to captive accounts), there is no need for that user to be notified of new mail. Thus, when you decide to prohibit mail access for a user, you might want to disable the new mail message at the same time. The following AUTHORIZE qualifier accomplishes both tasks:

```
/FLAGS=(DISMAIL,DISNEWMAIL)
```

5.2.7.5 Controlling Disconnected Jobs

Virtual terminals allow users to maintain more than one disconnected process at a time. You may want to restrict the use of virtual terminals. For example, if you become concerned about limited nonpaged pool, you may not want to enable this feature on a systemwide basis.

Virtual terminals can be disabled at the terminal or user level. To prevent particular terminals from being used as virtual terminals, use the DCL command SET TERMINAL/PERMANENT/DISCONNECT. To prevent specific users from attaching to disconnected processes, set the AUTHORIZE qualifier /FLAGS=DISRECONNECT for those users. (An applications account used by multiple users is a good candidate for the DISRECONNECT flag to prevent the users from connecting to each other's jobs.)

At the system level, you can disable virtual terminals on a systemwide basis with the SYSGEN parameter TTY_DEFCHAR2. However, this has other effects as well. You can also set the amount of time allowed for reconnection to less than the default of 15 minutes. Use the SYSGEN parameter TTY_TIMEOUT for this purpose, and the result also affects the system at large. Limiting the connection time tends to minimize the number of users who receive messages, but it also affects the usefulness of the connection feature.

5.2.7.6 Controlling the Number of Retries on Dialups

You can control the number of login attempts the user is allowed through a dialup line. If the user makes a typing mistake after obtaining the connection, the user does not automatically lose the connection. This option is useful for authorized users, while still restricting the number of unauthorized attempts.

To implement control of retries, use two of the *LGI parameters* provided with SYSGEN (see the *VMS System Generation Utility Manual*, LGI_RETRY_TMO and LGI_RETRY_LIM. If you do not change the parameters, the default values allow the users three retries with a 20-second interval between each. This means that users will lose the connection only if they fail to specify a valid password in three tries, or they spend more than 20 seconds between two of their tries.

Note that these values apply to every user on the system who is permitted to access the system through a dialup line.

The following example illustrates setting the total number of retry attempts to six, allowing a half-minute interval between tries. Since these LGI parameters are dynamic, you could change them and test them before performing the SYSGEN command WRITE CURRENT and rebooting the system.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  SET LGI_RETRY_LIM 6
SYSGEN>  SET LGI_RETRY_TMO 30
SYSGEN>  WRITE ACTIVE

{OPCOM messages show modification has been made}

SYSGEN>  EXIT
$
```

5.2.7.7 Controlling Break-In Detection and Evasion

Section 5.2.7.6 shows how to control the number of login retries for users dialing in. By limiting the number of retries to a reasonable number on each dialup login, you make the job of dialing up and trying every password combination more difficult for probing outsiders. This is insufficient to completely evade break-ins. First, an obstacle like redialing is not going to prove an effective deterrent. Second, this technique applies only to dialups.

The VMS operating system offers additional methods of discouraging break-in attempts. These methods also use SYSGEN parameters in the LGI category. One of the parameters (LGI_BRK_LIM) defines a threshold count for login failures. When the count of login failures exceeds the LGI_BRK_LIM value within a reasonable time interval, the system assumes a break-in is in progress. Only login failures caused by specifying invalid passwords are counted, and they must be from a specific source. That source can be any of the following combinations:

- A specific terminal and a specific valid user name. As described in a following section, you can override this default to count failures by user name only. Attempted logins using invalid user names never trigger break-in detection; however, they are counted together as a single class per terminal and are used to trigger security alarms. (See Section 5.9 for information about security alarms.)
- A specific remote node and a specific remote user name.
- The user name of the creator of a detached process.

By default, LGI_BRK_LIM permits five failed login attempts from one of these sources. (Security managers can adjust the value of LGI_BRK_LIM with SYSGEN.)

The SYSGEN parameter LGI_BRK_TERM controls the association of terminals and user names for counting failures. By default, VMS sets this parameter to 1 so that they are tracked together. If you set this parameter to 0, the terminal is not included in the association; the failures associate on user name only. This feature is useful if you use terminal servers, switches, or similar facilities in which the terminal name that VMS Login sees is not a good indication of the identity of the actual terminal.

Another key parameter, LGI_BRK_TMO, controls the time period in which login failures are detected and recorded. The initial failure on each source is given an expiration time that represents the current time plus the delta time given by LGI_BRK_TMO. Each additional failure on that source adds another delta of LGI_BRK_TMO to that entry, thus extending the length of time that break-in detection is in effect. The cumulative effect is that the more

failures made by a source, the greater the window of time in which additional failures will count toward the critical number defined by LGI_BRK_LIM. If no more failures occur by the time the expiration point is reached, all failures are forgiven for that source. Note, however, that the failure count is not reset by a successful login.

For example, assume the default values are in effect. LGI_BRK_LIM specifies no more than five login failures from one source. LGI_BRK_TMO is set for five minutes. Assume that an outsider starts sending user names and passwords to the system. When the first password fails, the clock starts to run and the user has four more tries in the next five minutes. When the second attempt fails about 30 seconds later, the user has three tries left that will be counted over the next 9.5 minutes. When the third attempt fails 30 seconds later, the login failure observation time extends to 14 minutes. The fourth failure occurs about one minute later; the fifth failure occurs within another 30 seconds. By this time, the observation time has reached 22.5 minutes. As a result, the next login failure from that source within 22.5 minutes will trigger evasive action.

The system tolerates an average rate of login failures that is the reciprocal of the parameter LGI_BRK_TMO. For example, if the default value of LGI_BRK_TMO (300 seconds, or five minutes) is in effect, the average rate of tolerable login failures is one every five minutes. When the rate of login failures exceeds the tolerable rate, and the critical number of five failures is reached (the default value of LGI_BRK_LIM), the system concludes a break-in is in progress and initiates evasive action.

The system stops accepting logins from the offending source for a period of time. When the source is a terminal (when LGI_BRK_TERM equals 1), for a period of time no one can log in from that terminal with the user name that is under suspicion. (However, other users may log in from that terminal.) A remote user triggering break-in evasion is prohibited from logging in from that node for a period of time. Consequently, login attempts that provide valid user name and password combinations that should otherwise succeed are rejected during this interval, but only from the presumed intruder at that source. Once the interval elapses, operations return to normal. As a result of this form of evasive action, outsiders are less likely to learn the correct password by using repetitive login attempts.

The duration of the evasive action is controlled by the LGI_HID_TIM parameter. The length of time depends on an additional random number (in the range of 1 to 1.5) used as a multiplier. The product of LGI_HID_TIM and the random number yields the actual duration of evasive action. The formula could be represented as follows:

```
Evasion time = LGI_HID_TIM * (random number)
```

The inclusion of a random amount of time helps obscure the true evasion time. An outsider who learned the value of LGI_HID_TIM could not be assured that the evasive action would persist for exactly that length of time.

These parameters affect all terminals, users, and nodes that access the system. Since they are dynamic, you can reset them without rebooting the system.

If the values of LGI_BRK_LIM and LGI_BRK_TMO can be learned or guessed, the outsider can attempt a system break-in over sufficiently long intervals that suspicion is not triggered. The outsider can also change terminals, nodes, and user names frequently enough to avoid detection. Do not rely on these break-in techniques as the sole means of security on your system.

The technique of counting failures per terminal and user name raises the potential for break-in because the password guess rate for a particular user name is multiplied by the number of available terminals. Each terminal is counted as a separate source for break-in detection. The benefit of this approach, however, is that it sharply reduces the denial of service problem that could result from simply counting failures per terminal or per user name. A malicious user could disable an entire terminal room or user's account for a period of time if failures are counted for each user name alone.

By setting LGI_BRK_TERM to 0, you can detect attempts more quickly, at the expense of increasing the risk of denial of service to legitimate users.

The SYSGEN parameter LGI_BRK_DISUSER makes the effects of break-in detection more severe. If you set this parameter to 1, VMS sets the DISUSER flag in the UAF record for the account where the break-in was attempted. Thus, that user name is disabled until you manually intervene. However, the service denial effects of this option can be very severe. A malicious user can put all known accounts, including yours, out of service in a short time. To recover, you must log in on the system console where the SYSTEM account is always allowed to log in. VMS stores information in the break-in database about login failures that originate from a specific source. Use the DCL command SHOW INTRUSION to display the contents of the break-in database and the DELETE/INTRUSION_RECORD command to remove entries from the break-in database. See the *VMS DCL Dictionary* for additional information about these commands. Entries in the break-in database have the following format:

```
Intrusion       Type      Count      Expiration       Source
```

The information provided in the fields in each entry is as follows:

Intrusion	Class of intrusion.
Type	Severity of intrusion as defined by the threshold count for login failures. The SYSGEN parameter, LGI_BRK_LIM, defines the threshold count.
Count	Number of login failures associated with a particular source.
Expiration	Absolute time when VMS stops keeping track of login failure. The SYSGEN parameter, LGI_BRK_TMO, controls this time.
Source	Origin of the login failure.

The information in the break-in database is controlled by the SYSGEN parameters in the LGI category.

5.2.7.8 Using the Secure Server

Section 3.1.3.8 describes password grabbers as a class of programs designed to steal passwords from unsuspecting users who log in to terminals left on. The VMS operating system provides a secure terminal server that stops any currently executing process prior to the start of a login at that terminal.

Invoke the secure server separately for each terminal with the following DCL command:

```
SET TERMINAL/PERMANENT/SECURE/DISCONNECT
```

The user must then press the BREAK key followed by the RETURN key to initiate a login. The login proceeds as usual.

If you apply the secure server to all terminals, you make the login procedure consistent throughout the site and avoid possible confusion. However, certain applications that may use the terminal as a communications line may need to use the BREAK key for their own purposes. This would be incompatible with the secure terminal server.

The secure server feature is also incompatible with autobaud handling. However, because autobauding is only necessary on modem terminals (switched or dialup terminals), the modem handling on such terminals performs the equivalent of secure server functions. For secure operation, set up the terminal characteristics as follows:

- For local terminals (direct wired) use the following SET TERMINAL qualifiers:

  /NOMODEM/SECURE/DISCONNECT/NOAUTOBAUD

- For switched terminals (data switch and dialup), use the following SET TERMINAL qualifiers:

  /MODEM/AUTOBAUD/NOSECURE/DISCONNECT

Specify /DIALUP if the terminal port is accessible through a telephone line or the equivalent regardless of the path (direct modem, data switch, concentrator, or public data network).

Always specify /DISCONNECT to ensure against password grabbers. To prevent disconnected jobs from filling up your system, set the SYSGEN parameter TTY_TIMEOUT to a low timeout value, which determines when disconnected processes are deleted.

If you decide to apply the secure server to individual terminals, include directly wired terminals located in public areas or remote, unsecured areas. Terminals never used for local or dialup logins are not subject to this security problem. Terminals closely supervised during logins may also not require this measure. (Remember to put the SET TERMINAL commands in SYS$MANAGER:SYSTARTUP.COM.)

5.2.8 Using the Automatic Login Facility

You can assign accounts to particular terminals to enable an *automatic login* feature. This feature permits users to log in without specifying a user name. VMS associates the user name with the terminal and maintains these assignments in the file SYS$SYSTEM:SYSALF.DAT, referred to as the *automatic login file* or *ALF*. Maintain this file with the command procedure SYS$MANAGER:ALFMAINT.COM.

The ALF consists of one record for each terminal on which automatic logins are enabled. Each record consists of two fields: the device name of the terminal followed by the user name of an account. The device names must be unique within the file. However, the same user name can occur in any number of records; that is, one account can be automatically logged in to an unlimited number of terminals.

A newly installed VMS system contains an ALF with all terminals except the console terminal set for automatic login to the USER account. To provide individual users with private accounts, use AUTHORIZE to create separate accounts for each individual. Disable automatic logins to the USER account by removing the account in AUTHORIZE or by changing the password.

Use the ALFMAINT command procedure to maintain the ALF. The following commands invoke ALFMAINT for the system ALF:

```
$ SET DEFAULT SYS$MANAGER
$ @ALFMAINT
```

The ALF is an indexed file and does not need to be purged. Back it up regularly after a modification.

5.2.8.1 Adding New Records

Respond to the "Terminal (ddcu)?" prompt with the physical name of a terminal (OPA0, TTAx, or TXAx). Respond to the "Username?" prompt with a valid user name. If you enter an invalid terminal name, you receive an error message and are prompted again. The user name is not checked for validity.

5.2.8.2 Modifying Records

Respond to the "Terminal (ddcu)?" prompt with the name of a terminal already in the ALF. Respond to the "Username?" prompt with the name of the new user to be associated with the terminal. Respond to the prompt "Do you want to change this record (Y/N)?" with the letter Y (uppercase or lowercase) to modify the record; any other response cancels the modification.

5.2.8.3 Deleting Records

Respond to the "Terminal (ddcu)?" prompt with the name of a terminal already in the ALF. Respond to the "User?" prompt by pressing the RETURN key. Respond to the "Do you want to delete this record (Y/N)?" prompt with the letter Y (uppercase or lowercase) to delete the record; any other response cancels the deletion.

5.2.8.4 Exiting from ALFMAINT

Respond to the "Terminal (ddcu)?" prompt by pressing the RETURN key or typing EXIT.

Restricting ALF Users

To force individuals at specific terminals to log in to an application program, create a separate new account for the application, and use the ALF to set up automatic logins to the new account from the desired account as a captive account.

To access the ALF, set your default directory to SYS$MANAGER, and invoke the command procedure SYS$MANAGER:ALFMAINT.COM. The following example sets up a series of terminals for automatic login to the captive INVENTORY account:

```
$ SET DEFAULT SYS$MANAGER
$ @ALFMAINT
Terminal (ddcu)? TTA0              ! All terminals
Username? INVENTORY                ! on automatic
Terminal (ddcu)? TTA1              ! login
Username? INVENTORY
Terminal (ddcu)? TTA2
Username? INVENTORY
Terminal (ddcu)? TTA3
Username? INVENTORY
Terminal (ddcu)? EXIT
```

5.2.8.5 Logging In to an Automatic Login Terminal

Once you set up a terminal for automatic login, it can only be used for the designated account. This is most useful for applications terminals planned for use by persons who may be unfamiliar with computers. Thus, an automatic login account will very likely also be a captive account.

The automatic login feature suppresses the user name prompt. All other login features (system password, primary and secondary passwords, and messages) function normally, if enabled.

Passwords are optional. If you want the account to be open to all users where the terminals are located, eliminate the password. When no password is required, the user has no data to enter at login. VMS logs the terminal in automatically in response to the BREAK key or the RETURN key and immediately enters the application if the account is captive.

5.2.8.6 Protecting Automatic Login Accounts

Automatic login accounts are potentially accessible from terminals and sources other than the terminals listed in the ALF and, therefore, require protection, especially if they have no password, as follows:

1. Restrict network and dialup access, as appropriate, with the AUTHORIZE qualifiers /NODIALUP, /NONETWORK, and /NOREMOTE.
2. Set the AUTOLOGIN flag in the account's UAF record. This flag makes the account available only by autologin, batch, and network proxy.

5.3 Authorizing Usage

As you authorize users, consider what restrictions should apply to each to provide additional control over their use. You can restrict them to certain devices, commands, privileges, short account durations, working times, and certain modes of operation (batch, dialup, remote, network, local, or interactive).

5.3.1 Restricting Devices

There are a number of ways you can restrict the devices available to users. You may want to limit use to particular devices, or you may want to limit amount of usage. The next sections describe the controls available to you for restricting the use of terminals, disk volumes, applications terminals, and miscellaneous devices.

5.3.1.1 Restricting Terminal Use

Through the SYSGEN parameters TTY_DEFPROT and TTY_OWNER, VMS sets up terminals to be accessible to the SYSTEM account only. (Users can log in because login always starts to execute as a system process.) To make terminals accessible to certain users as applications terminals, you may want to change any or all of the following:

- Protection
- Owner UIC
- ACLs

The application of system passwords limits the use of those terminals to users who know the system password. You can also incorporate SET PROTECTION/DEVICE and SET ACL/OBJECT=DEVICE commands for specific terminals (with appropriate protection codes) in the command procedure SYS$MANAGER:SYSTARTUP.COM.

5.3.1.2 Restricting Disk Volumes

Identify the user's default device and directory in the UAF record with the AUTHORIZE qualifiers /DEVICE and /DIRECTORY. You can limit the number of blocks available to the user on that disk (and any other disk) through the disk quota feature. (Use the VMS System Management (SYSMAN) Utility as described in the *VMS SYSMAN Utility Manual.*)

The volume protection in place on other disks controls how much access a user can obtain to the disks. The user's privileges, which can be extended or limited through the AUTHORIZE qualifier /PRIVILEGES, also influence the access available (see Section 5.3.6).

5.3.1.3 Applications Terminals and Miscellaneous Devices

Use the DCL command SET PROTECTION/DEVICE to limit access to any nonfile-structured device. You might also apply an access control list on the device to limit user access.

5.3.2 Restricting Work Times

AUTHORIZE qualifiers allow you to restrict system use to certain periods of the day. Define primary and secondary days of the week with the /PRIMEDAYS qualifier or conform to the default where primary days are Monday through Friday and secondary days are Saturday and Sunday. For example, if a user works Tuesday through Saturday, you would specify the /PRIMEDAYS qualifier as follows:

```
/PRIMEDAYS=(NOMON,TUES,WED,THUR,FRI,SAT,NOSUN)
```

Occasionally an operational change occurs that conflicts with the normal day assignments at your site, such as a holiday falling on a primary day. To override the normal day assignment, use the DCL command SET DAY, and specify the day-type interpretation you want for the current day. This requires OPER privilege. Note that this change applies to all logged-in users, as well as those who will log in during the day. Users who are currently logged in who are unauthorized for logins for the day-type once it changes will be logged off the system at the next hour. (The VMS Job Controller enforces time restrictions on an hourly basis.)

Decide which types of login access should be restricted to certain hours. The login access qualifiers are: /LOCAL, /REMOTE, /DIALUP, /INTERACTIVE, /BATCH, and /NETWORK. However, if your site applies one set of primary and secondary hours for all types of logins, you can specify the /ACCESS qualifier, which applies to all modes of access.

The following example shows how to apply the /BATCH qualifier to a user's account to disable the user from running batch jobs during normal working hours.

```
/NOBATCH=(PRIMARY, 9-17)
```

This specification permits the user to run batch jobs only during the hours of 6:00 p.m through 8:59 a.m. on primary days, but all day on secondary days.

You can use these to supervise users, or if you uncover evidence of damage occurring during a particular period. Restricting work times is also useful to better balance the workload on your system.

You may want to disable specific accounts used only periodically, such as the SYSTEST and FIELD accounts, to limit possible misuse of these accounts. Disable the accounts with the qualifier /FLAGS=DISUSER. Temporarily enable the accounts with the /FLAGS=NODISUSER qualifier when needed.

5.3.3 Restricting Mode of Operation

The following concerns might cause you to prohibit network access for some of your users:

- The user has data that should only be accessed through the local node.
- Penetration attempts are more likely to occur over a network because of the increased anonymity of the connection. (This concern is also relevant to dialup connections.)

Use the AUTHORIZE qualifier /NONETWORK to prevent specific users from having network access, as shown in the following example:

```
UAF> ADD JSMITH /NONETWORK, . . .
```

Any of the AUTHORIZE access mode qualifiers (/LOCAL, /REMOTE, /DIALUP, /INTERACTIVE, /BATCH, or /NETWORK) can be negated in this manner to restrict access to the system.

5.3.4 Restricting DCL Command Usage

There are several methods that you can apply to affect the use of DCL commands by your users. Among them are the following:

- Remove or modify DCL command definitions, and rebuild the DCL tables. (The *VMS Command Definition Utility Manual* describes how to create command definitions.) Use the /CLITABLES qualifier in the user's UAF record to specify the modified tables. Specify /FLAGS=DEFCLI to ensure that the user can only log in with the specified CLI and tables, and protect the original DCL tables from unauthorized access.
- Impose ACLs on the system program files in the directories SYS$SYSROOT:[SYSEXE] and SYS$SYSROOT:[SYSLIB].

5.3.5 Restricting Account Duration

It is good practice to set an account expiration time that matches the maximum length of time you expect the user to require access. When the expiration time arrives, the system automatically prohibits access to the account. You must still remove the UAF record and delete the user's files.

To set the account expiration time, use the AUTHORIZE qualifier /EXPIRATION in the user's UAF record. For example, the following qualifier specifies that the user's account will expire on the 30th of December, 1988:

```
/EXPIRATION=30-DEC-1988
```

5.3.6 Granting User Privileges

Some system activities are limited to users who hold specific privileges. These restrictions protect the integrity of the operating system's performance and, thus, the integrity of service provided to users. Grant privileges to each user on the basis of two factors: (1) whether the user has a legitimate need for the privilege, and (2) whether the user has the skill and experience to use the privilege without disrupting the system.

Privileges are divided into seven categories according to the damage that the user possessing them could cause the system:

- None—No privileges
- Normal—Minimum privileges to effectively use the system
- Group—Potential to interfere with members of the same group
- Devour—Potential to consume noncritical systemwide resources
- System—Potential to interfere with normal system operation
- File—Potential to compromise file security
- All—Potential to control the system

A user cannot execute an image that requires a privilege the user does not possess unless the image is installed as a known image with the privilege in question. (See the *VMS Install Utility Manual* for instructions on installing known images.) Execution of a known image with privileges grants those privileges to the user process executing the image for the duration of the image's execution. Thus, you should install user images with amplified privileges only after ensuring that the user needs the access and is unlikely to misuse it.

A user's privileges are recorded in the user's UAF record in two 64-bit privilege vectors. One vector stores the authorized privileges, and the other vector stores the default privileges. The default privileges are the subset of authorized privileges that a user process receives at login.

When a user logs in to the system, the user's privilege vector is stored in the header of the user's process. In this way, the user's privileges are passed on to the process created for the user. Users can use the DCL command SET PROCESS/PRIVILEGES to enable and disable privileges for which they are authorized. A user with the SETPRV privilege can enable any privilege.

Table 5–2 categorizes the privileges and includes a brief definition of the powers associated with each privilege.

Table 5–2 VMS Privileges

Category	Privilege	Activity Permitted
None	None	None requiring privileges
Normal	MOUNT	Execute mount volume QIO
	NETMBX	Create network connections
	TMPMBX	Create temporary mailbox
Group	GROUP	Control processes in the same group
	GRPPRV	Group access through SYSTEM protection field

Table 5–2 (Cont.) VMS Privileges

Category	Privilege	Activity Permitted
Devour	ACNT	Disable accounting
	ALLSPOOL	Allocate spooled devices
	BUGCHK	Make bugcheck error log entries
	EXQUOTA	Exceed disk quotas
	GRPNAM	Insert group logical names in the name table
	PRMCEB	Create/delete permanent common event flag clusters
	PRMGBL	Create permanent global sections
	PRMMBX	Create permanent mailboxes
	SHMEM	Create/delete structures in shared memory
System	ALTPRI	Set base priority higher than allotment
	OPER	Perform operator functions
	PSWAPM	Change process swap mode
	WORLD	Control any process
	SECURITY	Perform security related functions
	SYSLCK	Lock systemwide resources
Files	DIAGNOSE	Diagnose devices
	SYSGBL	Create systemwide global sections
	VOLPRO	Override volume protection
All	BYPASS	Disregard protection
	CMEXEC	Change to executive mode
	CMKRNL	Change to kernel mode
	DETACH	Create detached processes of arbitrary UIC
	LOG_IO	Issue logical I/O requests
	PFNMAP	Map to specific physical pages
	PHY_IO	Issue physical I/O requests
	READALL	Possess read access to everything
	SETPRV	Enable any privilege
	SHARE	Access devices allocated to other users
	SYSNAM	Insert system logical names in the name table
	SYSPRV	Access objects through SYSTEM protection field

5.3.6.1 Limiting User Privileges

Granting privileges allows users those privileges until you remove them. To avoid such blanket permission, you may want to grant privileges on an as-needed basis. For example, certain users might need to run a program requiring any of the more powerful privileges. You could install the program with the necessary privilege using the VMS Install Utility. Then put an ACL on the executable image file to clearly specify users allowed to execute it. The users would effectively possess the privilege only when they are actually executing the image. When the image stops running, the user no longer holds the privilege.

Following is an example of this method. The program SDA.EXE (required to analyze system dumps) requires CMKRNL privilege to analyze the running system. The security manager installs SDA.EXE with the CMKRNL privilege, as follows:

```
$ RUN SYS$SYSTEM:INSTALL
SDA.EXE /PRIVILEGED=CMKRNL
{messages}
```

Next, the security manager places an ACL on SDA.EXE and also sets the UIC-based protection to deny all access to the WORLD category of users, as follows:

```
$ SET ACL/ACL=(IDENTIFIER=SDAUSERS,ACCESS=EXECUTE) SYS$SYSTEM:SDA.EXE
$ SET PROTECTION=(WORLD) SYS$SYSTEM:SDA.EXE
```

Finally, the security manager uses the AUTHORIZE command SHOW/IDENTIFIER=SDAUSERS to confirm that the users who hold the SDAUSERS identifier are those intended to run the program. If necessary, the manager makes adjustments to this list of users.

DIGITAL ensures that all system programs (such as SDA) that are supplied with VMS are linked with the /NOTRACE qualifier to prevent online debugging or traceback. Take the same precaution with your own images. Online debugging and traceback are prime sources of security problems.

Note: All images that you install with privilege must be linked with the /NOTRACE qualifier to prevent online debugging and traceback.

Another alternative to granting blanket privileges is to set up emergency or specialized privileged accounts. Users would only log in to these privileged accounts to perform specific functions. You have two options with this technique. First, you establish a limited group who know about the account and are informed how to use it. Second, you create two accounts for the user, giving the privileges to one account but not to the other. In this case, the user would have the same UIC and the same default directory in each account. (This is the only case where DIGITAL recommends shared UICs, since there is still only one actual user.) If you decide to adopt this dual account practice, avoid obvious user names that reveal which account is the privileged account.

With both options, you can place special restrictions on the privileged account, such as long passwords, brief password lifetimes, restricted hours, and limited modes of operation (no dialup, no network, remote, or batch access). Also, limited account durations would force frequent consideration of privilege requirements.

5.3.6.2 Suggested Privilege Allocations

Appendix A lists all user privileges and includes recommendations on when to grant them. When allocating user privileges, consult Table 5–2. Be conservative.

The summary guidelines in Table 5–3 indicate the minimum privilege requirements for common classes of system users.

Table 5–3 Minimum Privileges for System Users

Type of User	Minimum Privileges
General	TMBMBX,NETMBX
Operator	OPER
Group Manager	GROUP,GRPPRV
System Manager/Administrator	SYSPRV,SETPRV
Security Manager	SETPRV,SECURITY

5.3.6.3 Controlling Privileged Accounts

Since abuse of privileged accounts can result in serious losses, consider imposing special controls on accounts with the most powerful privileges, as follows:

- Limit access to the account. For example, you can prohibit dialup or network access with the /NODIALUP or /NONETWORK qualifiers to discourage outsiders from attempting break-ins from remote locations.
- Use the /PRIMEDAYS and /NOACCESS qualifiers to restrict the time of day or days of the week that logins can be performed. Select periods of time that can be monitored for appropriate use.
- Disable the account when not in use with the AUTHORIZE qualifier /FLAGS=DISUSER.
- Use a captive login command procedure for additional validation. Captive login command procedures are described in Section 5.8.1.1.
- Impose security alarms to detect abuses of the privileges pertaining to file protection: BYPASS, SYSPRV, READALL, and GRPPRV. For information about setting up and monitoring security alarms, see Section 5.9.

5.3.6.4 Special Purpose Privileged Captive Accounts

Although generally unadvisable, it is sometimes necessary to grant privileges to captive accounts rather than giving users access to unrestricted, privileged accounts. For example, users who perform backup operations require the READALL privilege. By making the account that performs backups captive, you can ensure that the procedures are carried out according to your system's backup policy.

Guidelines in setting up captive accounts are provided in Section 5.8.

5.3.7 Examples of Establishing User Accounts

This section illustrates the creation of three user accounts with options ranging from least to most restrictive. Chapter 7 includes a similar example that illustrates a number of principles involved in designing and implementing proxy login accounts.

The first account represents highly privileged users, such as system managers, with minimum restrictions and maximum access to the system.

The second example illustrates an interactive user account with moderate restrictions, typical of an account at a commercial site where security is a concern and the average user has limited access.

The third example depicts an applications production account where the user is highly restricted.

In the following examples, any value not specified defaults to the value provided by the DEFAULT record in the UAF.

5.3.7.1 A System Manager's Account

Example 5–1 illustrates a number of AUTHORIZE qualifiers appropriate for a system manager's account. Notice the use of a short password lifetime, the requirement that the automatic password generator be used to change passwords, and the use of primary and secondary passwords. These measures are important to protect the account because it affords many valuable privileges and access rights.

Example 5–1 Example of a Security/System Manager's Account

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD RIRONWOOD/PASSWORD=(VALTERSY,ESTREMAY)/UIC=[001,100] -
_UAF> /DEVICE=SYS$SYSDEVICE/DIRECTORY=[RIRONWOOD] -
_UAF> /OWNER="Russ Ironwood"/ACCOUNT=SECURITY/FLAGS=GENPWD -
_UAF> /PWDLIFETIME=30-/PWDMINIMUM=8 -
_UAF> /PRIVILEGES=SETPRV
identifier for value:[000001,000100] added to RIGHTSLIST.DAT
UAF>
```

This user has the most powerful privilege of all, SETPRV. With this privilege the owner can acquire any other privilege.

5.3.7.2 A Typical Interactive User's Account

Example 5–2 illustrates the creation of an account of a typical interactive user. Only one password is required, with a minimum length of 6 characters. The user's password is valid for 90 days, a much longer lifetime than the manager's password. The user is allowed access during the week and on Saturdays, during a fifteen-hour period.

Example 5–2 Example of a Typical Interactive User Account

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD RDOGWOOD /PASSWORD=TRALAYAM/UIC=[231,010] -
_UAF> /DEVICE=BOTANYDEV/DIRECTORY=[RDOGWOOD] -
_UAF> /OWNER="Robert Dogwood"/ACCOUNT=BOTNYDPT -
_UAF> /FLAGS=(GENPWD) -
_UAF> /PRIMEDAYS=(MON,TUES,WED,THURS,FRI,SAT,NOSUN) -
_UAF> /EXPIRATION=15-JUNE-1987/PWDLIFETIME=90-/PWDMINIMUM=6 -
_UAF> /NOACCESS=(PRIMARY,23-6,SECONDARY)/NODIALUP
identifier for value:[000231,000010] added to RIGHTSLIST.DAT
UAF>
```

5.3.7.3 A Production Account

Example 5–3 illustrates the creation of a production account. This account is designed to perform one function: to list the grades at State University and to produce mailings to each student's home. This job can only be run from the captive account REPGRADES, and it may not be run over dialup lines or as a remote job. Nor will the account permit network access. When the job is run through a local login, it is restricted to the hours of 8 a.m. through 5:59 p.m., Monday through Friday. However, when the job is run in batch mode, it is not restricted to special times. The user who initiates the login must specify the password, GROBWACH. (Most likely only the security manager will change the password.) The process runs under the control of a special login command procedure (GRADES.COM), which presumably provides the operator with a menu of functions. (The process is restricted to the commands defined in the CLI table, GRADES_TABLES.)

Example 5–3 Example of a Production Account

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD REPGRADES /PASSWORD="GROBWACH"/UIC=[777,031] -
_UAF> /DEVICE=ADMINDEV/DIRECTORY=[REPGRADES] -
_UAF> /OWNER="Campus Admin"/ACCOUNT=ADMIN -
_UAF> /FLAGS=(CAPTIVE,DISWELCOME,DISNEWMAIL,DISMAIL,DEFCLI) -
_UAF> /LOCAL=(PRIMARY, 8-17)/PRIMEDAYS=(MON,TUES,WED,THU,FRI,NOSAT,NOSUN) -
_UAF> /NONETWORK/NOREMOTE/NODIALUP -
_UAF> /LGICMD=GRADES/CLITABLES=GRADES_TABLES -
_UAF>
   .
   .
   .
user record successfully added
identifier for value:[000777,000031] added to RIGHTSLIST.DAT
```

5.3.8 Training the New User

Teaching new users about system security is an important security tool. It is important to involve users in security methods and goals; the more they know about the system and how break-ins occur, the better equipped they are to guard against them.

Following is a list of topics to include in user training:

- What is the location of the user's account? Specifically, which system, where is it located, what is the proper node name if on a network, and if the system is part of a cluster, what other nodes are available?
- Which terminals can be used for logging in and where are they located?
- Is the account restricted with regard to local, dialup, remote, interactive, network, or batch operations? If so, describe both permitted use and restrictions.
- Can the account be accessed by dialing in? If so, provide the access telephone number and describe the procedure. Specify how many retries are allowed and the maximum number of seconds allowed between each before the connection is lost.

- Are system passwords implemented for any terminals that the user may be using? If so, describe which terminals, how often the system password is changed, and how the user can learn the new system password.
- What is the account duration? When will it expire? From whom should the user request an extension?
- What is the user name? What identifiers are held by the user, if any? What are the group and member numbers associated with the user?
- What password information is required? Specifically, what is the initial password? Is the password locked? If the password is not locked, how often must the password be changed? What is the minimum length for the password? Is there a secondary password for this account, and who will know it? Is the user free to select passwords, or must they be automatically generated?
- What is the default device and directory?
- What is the default protection?
- Are there quotas on disk usage? If so, what are the values?
- Are there restrictions on use? For example, are there certain days or hours of the day that are suggested or enforced? Explain primary and secondary days if applicable.
- Are there files or directories that are shared? If so, provide the details.
- Are there ACLs that affect the user? What identifiers does the user need to know?
- Which privileges does the user hold?
- What is the command language interface?
- Which type of account is this—open, locked, captive, or normal?
- Which nodes permit proxy logins for this user, if any?
- What are the names of the queues the user may need to use?
- Describe actions related to the physical aspects of security, such as locking up materials.

5.4 Protecting Information

After designing an appropriate system access plan, you must address which files you want your users to access. Even on the most open system, you will want protection for the system software. DIGITAL provides default UIC protection for this purpose that is generally sufficient. However, you may want to override the default UIC protection or set up ACLs for specific system files. Table 5–4 provides a summary of DCL commands you use to set up and display file protection.

Table 5–4 DCL Commands Used to Protect Files

Command	Function
DIRECTORY/ACL	Displays the ACL for the file
DIRECTORY/OWNER	Displays the file owner's UIC
DIRECTORY/PROTECTION	Displays the file's protection mask
DIRECTORY/SECURITY	Combines and displays file information produced by DIRECTORY/ACL, DIRECTORY/OWNER,and DIRECTORY/PROTECTION
EDIT/ACL	Invokes the Access Control List (ACL) Editor
SET ACL	Creates, modifies, or deletes access control list entries (ACEs) in an ACL
SET ACL/EDIT	Invokes the Access Control List (ACL) Editor (synonymous with EDIT/ACL)
SET DIRECTORY/OWNER_UIC	Modifies the owner UIC of the directory
SET FILE/OWNER_UIC	Modifies the owner UIC of the file
SET FILE/PROTECTION	Changes the file's protection code
SET PROTECTION	Changes the file's protection code (synonymous with SET FILE/PROTECTION)
SHOW ACL	Displays the file's ACL

These commands are described in the *VMS DCL Dictionary*.

5.4.1 Restricting Command Outputs

Some DCL commands behave differently depending on the privileges that the user holds.

For example, unless a user holds the GROUP or WORLD privilege, the SHOW PROCESS command limits the display of process information to the user's process. A user with GROUP privilege can display other processes in the user's UIC group; a user with WORLD privilege can display any process on the system.

5.4.2 Protecting System Programs and Databases

Normally, DIGITAL delivers system programs and databases with adequate protection. However, if for any reason you are dissatisfied with the default protection, you can change it with the techniques outlined in Chapter 4, provided you have the necessary SYSPRV privilege. You might also add an ACL to any file that you decide needs additional protection. Appendix C presents the recommended protection codes for all system files. Your VMS software should have this set of protection codes following a correct installation. Examine your system files from time to time to ensure that this protection is maintained.

As indicated, DIGITAL provides default protection for the system programs that it provides. However, if you have a special requirement, you might examine the potential of ACLs for your needs. For example, you could consider using ACLs to restrict the use of system programs such as compilers. (Any number of considerations might prompt this action, ranging from performance to licensing issues.)

Another possible question worth investigating is: are there cases where you do not want some or all of your users able to initialize media? If so, you could put an ACL to good use on the system program SYS$SYSTEM:INIT.EXE. You would ensure that you grant no access to the WORLD category in the UIC-based protection field. Then create an ACL for the file that grants access to the specific users you want.

Similarly, if a department in your company has paid for a license to a software product, you may want to make that software available to them, but not to others. You would ensure that the WORLD category receives no access through the standard UIC-based protection code and create an ACE in the ACL for that file that allows the access through the department's identifier.

You may also find that ACL protection is relevant to protect your applications databases, limiting the access to certain users.

5.4.3 Precautions to Take When Installing New Software

When you install new software, you must address several security concerns. You want to ensure that you are not admitting software that will in any way corrupt or undermine your usual security precautions. You must also consider whether or not to install the software with any privileges. When you install privileged software, you allow users to execute it whether or not they personally possess the required privilege. In effect, you extend the privilege to the process while it runs the software. While this offers some advantages with clear security implications, it also introduces several security-related dangers. This section discusses security aspects of installing new software.

5.4.3.1 Protecting Programs and Directories

New software can contain programs that are potentially harmful to your system. These programs, called *Trojan horse* programs, are designed to do damage and frequently include devices that do the following:

- Pass privileges of the person running the program back to the author of the program
- Allow unauthorized access to the system
- Change protection of system files
- Patch the system (add special software to the operating system)

To protect your system from this type of break-in, always buy software from reputable sources. When training new users, stress the importance of avoiding use of software from an unknown source.

Another risk to programs and directories is known as the *worm*. While Trojan horse software must rely on the innocent user to unwittingly accept the damaging software by using it, the worm requires no user cooperation. It is a program that takes advantage of faulty file protection, working its way through your system modifying command procedures and executable

programs. By modifying command procedures, it can propagate by making use of user access rights and privileges.

The user's login command procedure is a prime target for this type of security breach. Login command procedures generally contain easily modified DCL commands and are executed regularly.

ACLs are also targets. File protection designed with users sharing access privileges allows this type of program to run through many users' programs, acquiring new privileges along the way.

Well-designed file protection is vital for protection from this type of breach. Make sure that likely targets are not modifiable by users. For example, set up file protection so that your login command procedure permits only READ access to all other users. Also, make sure the directory containing the login command procedure permits WRITE access only to users in the SYSTEM and and OWNER categories.

Because most damage occurs when programs like these reach a target account with privileges, users with privileges should be especially cautious when designing file protection.

5.4.3.2 Installing Programs with Privilege

Some software requires privilege to run. You can extend the privilege to all users you expect will need to run the software, or install the program with the required privileges. Section 5.3.6 describes these options in greater detail.

5.5 File Encryption

File encryption refers to the process of applying an algorithm to data to conceal its content. *Decryption* reverses the operation and converts encoded information back to its original content. If you needed to copy proprietary software onto media for removal to another site, you might use file encryption. The software on the media is useless without the correct decryption code.

To perform these tasks, there is a software facility available to VMS users as an optional layered product. Consult the *VMS Data Encryption Facility Manual* for more information.

5.6 Disk Maintenance Considerations

Proper disk maintainence includes the following:

- Physical security for disks
- Backups of disks
- Physical security for backups
- How to retrieve files from backups

Having an effective backup system is vital to protect your data. Without a backup system, recovery of deleted or damaged files is impossible. Never give a copy of your backup media to a user; a malicious user could restore the files from the tape or disk and compromise the security of the system. Refer to the *VMS Backup Utility Manual* for information about setting up backup systems.

5.7 Methods for Discouraging Disk Scavenging

Disk scavenging is the process of reading magnetic imprints of data after deletion of the file header following a purge or delete operation. (When users delete files from the system, only the file header is deleted.) Until the data is overwritten, it is a potential target for disk scavenging. Sites with medium or high security needs should be concerned about this procedure.

After establishing overall security features, restrict access to disks containing valuable information using UIC-based volume protection. Because disk scavenging is frequently performed by authorized users, consider implementing erasure patterns and highwater marking, as described in the following sections.

5.7.1 Erasing Techniques

There are several ways to implement erasing of disks.

The inclusion of the /ERASE qualifier with the DELETE or PURGE commands causes the system to write an erasure pattern of zeros over the entire file location when you delete or purge that file. You can encourage users to use this qualifier voluntarily, or make inclusion automatic by including the following command definitions in the system login command procedure (usually SYS$MANAGER:SYLOGIN.COM):

```
DEL*ETE :== "DELETE/ERASE"
PUR*GE :== "PURGE/ERASE"
```

Any user can bypass these definitions by adding the /NOERASE qualifier to the DELETE or PURGE commands.

To guarantee *erase-on-delete,* turn on the feature for the entire volume using the DCL command SET VOLUME/ERASE_ON_DELETE. When files are deleted, this command overwrites all files on the volume with the erasure pattern of zeros.

To completely erase the volume and enable erase-on-delete for the volume at volume initialization, use the DCL command INITIALIZE/ERASE.

By default, VMS writes a default *data security erase (DSE)* pattern of zeros, applied during a single WRITE operation over the area, when erase-on-delete is enabled. If you feel that the default pattern of zeros or the single rather than multiple number of erasures does not suit your requirements, you can use the $ERAPAT (Get Security Erase Pattern) system service to write a customized erasure pattern to specified files. See the description of $ERAPAT in the *VMS System Services Reference Manual* for more information.

Generally, for sites with high-level security requirements, a random pattern is preferable to a fixed pattern. The technology is already available that can detect and use faint residual magnetic impressions. Thus, if you conclude there is sufficient danger that a disk might be removed and read by some of this specialized analysis equipment, you might need to rewrite the erasure pattern several times. You can learn how to customize the data security erase pattern to fit your needs by studying the information provided in the file SYS$EXAMPLES:DOD_ERAPAT.MAR.

Employ erasing patterns only on disks where the security needs are the greatest. Erasures are time-consuming and affect system performance.

5.7.2 Prevention Through Highwater Marking

Section 4.7 introduces the concept of highwater marking. Highwater marking refers to a technique that tracks the furthest extent to which each file has been written and prohibits user attempts at reading data beyond that point.

The VMS operating system implements true highwater marking for all sequential, exclusively-accessed files, such the set of files output from various text editors, compilers, and linkers; that is, most files a process writes. The highwater mark is updated in the file header whenever the logical end-of-file mark is updated (usually when the file is closed).

For indexed as well as sequential, shared files, the VMS operating system uses the principle of *erase-on-allocate* to achieve a result similiar to true highwater marking. When a file is about to be created or extended, VMS determines how much disk space (the extent of the file) is required and applies the security erasure pattern of zeros to the areas (extents) it allocates for writing. The file is then written into the area just erased for it. Thus, if any user gains access to the file (including its full extent) and attempts to read the area beyond where the file has been written, only the data security erase pattern is readable.

By default, VMS turns on highwater marking for all volumes. Highwater marking is a deterrent to disk scavenging attempts. However, it does require additional I/O, which affects system performance.

You can turn off highwater marking on a volume-by-volume basis by specifying the DCL command SET VOLUME/NOHIGHWATER.

5.7.3 Summary of Prevention Techniques

Security managers can apply the following controls to discourage disk scavengers:

- Provide tight physical security, particularly on those disks with the most valuable information
- Provide tight volume protection through UIC-based protection
- Encourage the use of the /ERASE qualifier when key files are purged or deleted through user participation or volume enforcement
- Permit default highwater marking on your most valuable disks

5.8 Restricting the Environment—Captive Accounts

Following are situations where a restricted environment is advantageous:

- Permitting unskilled or semi-skilled users to perform routine computer tasks
- Running batch operations during unsupervised periods
- Running applications programs with information that you want to keep private

These situations can be best accommodated through a captive account. A captive account, also called a turnkey account, allows limited access to the system, usually through a specialized login command procedure. An example of a captive account appears in Section 5.3.7.3. However, this section will describe captive accounts in more detail, including two subcategories of captive accounts, guest accounts and proxy login accounts.

5.8.1 Creating a Captive Account

A captive account limits the activities of the user and usually denies the user access to the DCL command level. You can set up the account to restrict the user to running a specific program or command procedure. Define a captive account with AUTHORIZE by including the following qualifier when creating the account:

```
/FLAGS=(CAPTIVE)
```

This flag ensures that the account is noted as captive and disables CTRL/Y interrupts. Setting the CAPTIVE flag also prohibits the user from specifying an alternate command language interpreter (CLI) at login with the /CLI qualifier.

You may want to disable the welcome announcement and electronic mail for the captive account. This is done by setting the DISWELCOME, DISMAIL, and DISNEWMAIL login flags. Because the VMS Mail Utility has the ability to spawn other processes, you should prohibit captive account users from accessing electronic mail, or set the AUTHORIZE qualifier /PRCLM (subprocess limit) to 0 to prevent the user from spawning out of the captive account.

Decide whether users should be able to change the password for the captive account. There are two password options appropriate with captive accounts:

- Require no password by specifying the AUTHORIZE qualifier /NOPASSWORD.
- Lock the password with the /FLAGS=LOCKPWD qualifier so that only the security manager can change it.

Locked passwords are generally preferable to open captive accounts (those with no password). If you assign a locked password, give that password to all users of the captive account.

Your application may require you to impose additional AUTHORIZE qualifiers on the account, such as /NODIALUP to restrict modes of operation. Consider imposing restrictions for the periods of the day and days of the week when the process can run.

A captive account user is usually denied all access to the DCL command level. You can define a special set of DCL tables using the /CLITABLES qualifier, or you can emulate DCL through the use of a DCL command procedure. While using a restricted set of DCL tables will make it difficult for a user to exceed the intended limitations of the environment, this technique is not really secure. The sophisticated user may invoke the foreign command feature of DCL, which may only be suppressed by unsupported patches to DCL. It is, however, more efficient to define DCL tables than to resort to a DCL command procedure to emulate DCL. See the description of the VMS Command Definition Utility in the *VMS Command Definition Utility Manual* for help in defining the DCL tables.

If you define an alternate DCL command table for captive accounts, prevent captive account users from regaining access to the full DCL command set by denying WORLD access to SYS$LIBRARY:DCLTABLES.EXE.

You should rarely need to grant any privilege other than TMPMBX to a captive account.

Limit the disk quota for the captive account to the amount needed. A minimum diskquota of 300 to 500 blocks is recommended.

5.8.1.1 Login Command File Considerations

The primary feature of the captive account is its login command procedure. To override the default login command file (LOGIN.COM in the user's default directory), identify the captive account login command procedure with the AUTHORIZE qualifier /LGICMD. Use the following guidelines to make certain that the login command procedure is tamperproof:

- If the captive account user is allowed to create or perform other operations on files, make certain that WRITE access to the login command procedure and its directory is denied. (EXECUTE access is required.)
- Ensure that the account is not a member of the SYSTEM group (has a group value less than 10 octal, unless modified by the SYSGEN parameter MAXSYSGROUP).
- Give the captive account a unique UIC group. Use the following form of the AUTHORIZE command SHOW to verify this:

  ```
  SHOW [groupuic,*]
  ```

  By keeping the account in a separate group, you can ensure that the captive users can only access WORLD-accessible files and files owned by the captive account.

To prevent the user from escaping from the command procedure, and because the main source of escape is through the DCL command CTRL/Y, the general strategy is to use the CAPTIVE flag to disable CRTL/Y for the account. There are two types of captive applications where it is appropriate to allow the user to enter the CTRL/Y command subsequent to the startup of the command procedure.

- You may want to provide users with a CTRL/Y feature at points during the execution of the captive login command procedure. Include ON CONTROL_Y commands in the procedure where you want to test for CTRL/Y, as shown in Example 5–4.
- You may have a captive command procedure that ultimately turns control over to the user. For example, consider a SYLOGIN.COM command procedure that performs additional security validation; its execution should be guaranteed to ensure its effectiveness. However, once SYLOGIN.COM has done its job, control can be turned over to the user. To achieve this arrangement, mark the account as captive and enter the DCL command SET CONTROL=Y when you are ready to release control to the user, as shown in Example 5–5.

Do not allow the DCL command INQUIRE to appear in any command procedures. The INQUIRE command performs an evaluation while receiving input, which could create an opening to break the captive account. Instead, use the DCL command READ/PROMPT.

For example, to request the user to enter the date, enter the following command:

```
READ/PROMPT="Enter date: " SYS$COMMAND DATE
```

For similar reasons, avoid any use of the construction *'x*, where *x* contains a string entered by the user. Never permit a captive command procedure to attempt an evaluation of a symbol that the user enters. Use of lexical functions could break the command procedure.

If the function of the command procedure requires text preparation, you may want to give users access to an editor. When designing this environment, remember that most editors are capable of reading and writing files (within the access rights of the account).

Note: Do not permit the captive account command procedure to use the TECO editor because it can break out of any captive command procedure.

Example 5–4 shows a command procedure that provides a completely captive environment, restricting the user to a limited set of commands. Note that the security manager would use the AUTHORIZE qualifier /NOINTERACTIVE when establishing this account.

The sample captive command file in Example 5–5 can be used on privileged accounts. To allow only interactive use of the account from a local terminal, include the AUTHORIZE qualifiers /NODIALUP, /NOREMOTE, /NOBATCH, and /NONETWORK when establishing the account.

5.8.1.2 Guest Accounts

Guest accounts allow remote users access to resources on your system. For example, users across the network may need access to your system to report problems or read corporate memos.

DIGITAL does not recommend the practice of setting up guest accounts. Guest accounts, however unprivileged, offer malicious users a chance to compromise your system security. Most needs for a guest account can be handled by special proxy login accounts, which should also be captive accounts.

If you still need a guest account, take the following steps to make the account secure:

- Use an obscure password for the guest account. Change the password frequently. Never use easily guessable account name and password combinations such as GUEST/GUEST or USER/USER.
- Maintain a list of people allowed to use the account. (Changing the password regularly will help you keep this list current.)
- Set up the guest account in a separate UIC group. Make sure that the account is not a member of the SYSTEM group.
- Place the default login command procedure in the directory SYS$MANAGER using the AUTHORIZE command MODIFY, as follows:

  ```
  MODIFY guest-account/LGICMD=SYS$MANAGER:filename.COM
  ```

- Make the guest account captive by setting the AUTHORIZE qualifier /FLAGS, as follows:

  ```
  /FLAGS=(DISCTLY,LOCKPWD,CAPTIVE)
  ```

- Limit the number of subprocesses that the account can create to 0 using the AUTHORIZE qualifier /PRCLM=0.

Example 5–4 Example of a Captive Command Procedure

```
$ deassign sys$input
$ prev_sysinput == f$logical("SYS$INPUT")
$ on control_y then $goto next_cmd
$ set control=(y,t)
$next_cmd:
$ on error then $goto next_cmd
$ if prev_sysinput .nes. f$logical("SYS$INPUT") then deassign sys$input
$ read /end=next_cmd /prompt="$ " sys$command cmd
$ cmd :== 'cmd
$!
$ delete = "delete"
$ delete /symbol /local /all
$ if f$locate ("@", cmd) .ne. f$length(cmd) then goto illegal_cmd
$ if f$locate ("=", cmd) .ne. f$length(cmd) then goto illegal_cmd
$ t1 = f$locate (" ",cmd)
$ cmd1 = f$extract (0, t1, cmd)
$ if f$locate (cmd1, "LOGOUT") .eq. 0 then goto logout
$ if f$locate (cmd1, "HELP") .eq. 0 then goto help
$!
$! Place other validation checks here
$!
$ write sys$output "%CAPTIVE-W-IVVERB, unrecognized command"
$ write sys$output "   \",cmd1,"\"
$ goto next_cmd
$!
$illegal_cmd:
$ write sys$output "%CAPTIVE-W-ILLEGAL, bad characters in command"
$ goto next_cmd
$!
$cmd_ok:
$ define sys$input sys$command:
$logout:
$ logout
$ goto next_cmd
$help:
$ help
$ goto next_cmd
$!
$! Place other prevalidated commands here
$!
```

Example 5–5 Sample Captive Procedure for Privileged Accounts

```
$ if f$mode() .nes. "INTERACTIVE" then $logout
$ term = f$logical("SYS$COMMAND")
$ if f$locate("_T", term) .eq. 0 then $goto allow
$ if f$locate("_OP",term) .ne. 0 then $logout
$allow:
$ set control=(y,t)
```

- Assign the guest account only TMPMBX privilege.
- To handle error conditions, include the following commands in the default login command procedure:

```
SET ON
SET NOCONTROLY
ON ERROR THEN LOGOUT/BRIEF
```

- If LOGOUT is defined as a global symbol and points to a command procedure (enter the DCL command SHOW SYMBOL LOGOUT to confirm this), include the following DCL command in the default login command procedure for the account:

```
DELETE/SYMBOL LOGOUT/GLOBAL
```

  This will eliminate the possibility the user could break the captive account at logout time by typing CTRL/Y.
- To prevent outsiders from misusing your system resources through the submission of batch jobs under the guest account, include the AUTHORIZE qualifier /NOBATCH when you create the account.
- Limit the disk quota for the guest account UIC to the amount needed.
- Do not allow the DCL command INQUIRE to appear in any of the command procedures.

5.8.1.3 Proxy Login Accounts

Generally, proxy login accounts should be set up as captive accounts. Proxy login accounts permit remote users to access a local account without specifying a password. Section 7.6.1.2 describes proxy login accounts. Note that many recommendations are the same as those for captive accounts.

5.9 Auditing with Security Alarms

Security alarms are messages sent to the security operator's terminal indicating specific events. Alarms can help you detect outsiders' attempts to break into the system and can be used to monitor undesirable activity at your site. For example, you might enable an alarm that sends a message to the security operator's terminal whenever a UAF record changes.

When dealing with security alarms, carefully select and enable the events to be audited, enable a security operator terminal, and monitor and make use of the alarm information.

5.9.1 Enabling Security Alarms

To enable security auditing, specify the DCL command SET AUDIT in the following format:

```
SET AUDIT /ALARM /ENABLE=keyword[,...]
```

Select the events to be audited by specifying one or more of the following keywords to the /ENABLE qualifier:

- ACL—Event requested by an ACL on a file or global section
- ALL—All possible events

- AUDIT—Execution of the SET AUDIT command
- AUTHORIZATION—Modifications to the system UAF file, network proxy authorization file, rights database, or changes to system and user passwords
- BREAKIN—Successful break-in attempt
- FILE_ACCESS—Selected types of access (privileged and nonprivileged) to files and global sections
- INSTALL—Installation of images
- LOGFAILURE—Failed login attempt
- LOGIN—Successful login attempt
- LOGOUT—Logout
- MOUNT—Volume mounts and dismounts

See the *VMS DCL Dictionary* for more information about the SET AUDIT command.

If you enable alarms for too many events, you diminish their effectiveness through overuse. Implement alarms only for key events.

See Section 6.2.2 for suggestions about which events you should enable if you suspect your system is under attack.

5.9.2 Enabling a Security Operator Terminal

Before you enable alarms for particular events, enable a security operator's terminal. Choose a terminal that provides hardcopy output and is located in a secure location. The following DCL command enables the terminal from which the command is entered:

```
$ REPLY/ENABLE=SECURITY
```

Any number of terminals can be enabled as security operators. If you designate one terminal as the security operator's terminal, add the following lines to the site-specific startup command procedure (usually SYS$MANAGER:SYSTARTUP.COM) to send alarms to the terminal and disable them on the system console:

```
$ DEFINE/USER SYS$COMMAND OPA0:
$ REPLY/DISABLE=SECURITY
$ DEFINE/USER SYS$COMMAND TTA3:
$ REPLY/ENABLE=SECURITY
```

Security alarms are always written to the operator log file, even if no security operator terminal is enabled.

5.9.3 Enabling Alarm Messages

After you enable a security operator terminal, enable specific alarm events with the SET AUDIT/ENABLE command. Alarm messages are then sent to the security operator terminal when the selected events occur. Security alarms appear as follows:

```
%%%%%%%%%%  OPCOM  30-DEC-1988 12:27:52.26  %%%%%%%%%% ❶
Security alarm on LASSIE / System UAF record modification ❷
       Time:           30-DEC-1988 12:27:52.25 ❸
       PID:            23C00155 ❹
       User Name:      MENACE ❹
       Rec Mod:        GOWER
       Fields Mod:     PRIVILEGES
```

The information included in the message depends on the type of event; in all cases, the alarm message contains the following four elements:

❶ OPCOM heading, which includes the date and time the alarm was sent

❷ Type of alarm event

❸ Date and time the alarm event occurred

❹ The user who caused the event, as identified by the user name and process identification (PID)

Other information contained in alarm messages is specific to the type of event that the alarm signaled. Appendix E includes examples of the alarm messages associated with particular alarm events.

5.9.4 Audit Reduction Facility

If you have enabled security alarms, the operating system writes information resulting from those alarms to the security operator's log file. Because you can enable alarms for many objects and types of access, the log file often contains a large volume of information. To selectively extract information from the operator's log file, use SECAUDIT.COM, a command procedure residing in SYS$MANAGER.

To extract all the security alarm information from the current operator's log file (SYS$MANAGER:OPERATOR.LOG), execute the following command:

```
$ @SYS$MANAGER:SECAUDIT
```

Output from SECAUDIT is displayed on SYS$OUTPUT. If you want to write the records to a file, include the file specification with the /OUTPUT qualifier. The following command writes the records to the file BREAKINS.DAT in the user's current default directory:

```
$ @SYS$MANAGER:SECAUDIT/OUTPUT=BREAKINS.DAT
```

5.9.4.1 Optional Parameters

SECAUDIT.COM accepts the following five optional positional parameters:

p1 Name of the log file that is scanned for the selected security alarm information. By default, SYS$MANAGER:OPERATOR.LOG is searched.

p2 Specific user name for which the relevant security alarms are to be displayed.

p3 Starting date and time of the first security alarm entry to be displayed.

p4 Ending date and time of the last security alarm entry to be displayed.

p5 Selection criteria - specify the selection criteria with the keywords used with the /ENABLE qualifier of the SET AUDIT command.

By default, SECAUDIT.COM searches SYS$MANAGER:OPERATOR.LOG for security alarm information. Use the p1 parameter to specify a log file that is different from the default.

The remaining parameters select specific alarm information. Using these parameters, you can select alarm information generated in the following ways:

- By specific users
- Within a particular time frame
- By particular types of alarms

For example, the following command extracts all security alarm records generated by the user SARDONO after February 1, 1988:

```
$ @SYS$MANAGER:SECAUDIT "" SARDONO 1-FEB-1988
```

Note that because the parameters are positional, you pass a null parameter by using a set of quotation marks as a placeholder in the command string.

Use the p5 parameter to select alarm information that resulted from a particular type of event. The operating system audits a number of events that are enabled by specifying keywords with the /ENABLE qualifier of the SET AUDIT command. Use the same keywords to select the alarm information from the operator's log.

For example, the following command extracts all security alarm records generated by break-in attempts, any access to a file using the SYSPRV privilege, or any access to a file using the BYPASS privilege:

```
$ @SYS$MANAGER:SECAUDIT "" "" "" "" BREAKIN,FILE_ACCESS=(SYSPRV,BYPASS)
```

5.9.5 Auditing a Terminal Session

You may need to audit an entire terminal session. If you set host to your own system and specify that a log file of the session be kept, the resulting log file will contain a record of the entire terminal session. For example, the following command keeps a record of the entire session in the log file APRIL15.LOG:

```
$ SET HOST 0 /LOG=APRIL15.LOG
```

Note that if you use the /LOG qualifier without including a file specification, the log information is stored in the file SETHOST.LOG.

Use of the SET HOST command requires that DECnet be configured and started. The installation does not have to buy a DECnet license; the DECnet license is only necessary to actually communicate with remote nodes. To use SET HOST in an environment without DECnet, you must configure a network database with no communications lines or remote nodes, and you must start the network. In the absence of supported communications equipment, NETCONFIG.COM will perform these steps correctly. Details on these operations are provided in the *VMS Networking Manual*.

5.9.6 Enforcing a Terminal Session Audit

You can enforce auditing of terminal sessions for selected users by use of a special captive account and appropriate command procedures. Users for whom session auditing is enforced must first log in to the special captive account and then in to their own account. The captive account assures that the session is audited. This section provides guidelines on how to set up the captive account (named USER_AUDIT in this example) and includes samples of appropriate command procedures. The captive account USER_AUDIT is set up as follows:

```
UAF> ADD USER_AUDIT /FLAGS=(CAPTIVE,DISMAIL,DISNEWMAIL) -
     /LGICMD=SYS$SYSROOT:[USER_AUDIT]AUDITLOG -
     /DEV=SYS$SYSROOT: /DIR=[USER_AUDIT] -
     /NONETWORK /NOBATCH /UIC=xxx ...
```

The AUDITLOG.COM command procedure enables auditing of the terminal session, as follows:

```
$ ! AUDITLOG.COM - log into specified account with terminal session
$ ! auditing enabled.
$ !
$ WRITE SYS$OUTPUT "Please log into the account of your choice."
$ WRITE SYS$OUTPUT "Your terminal session will be recorded."
$ WRITE SYS$OUTPUT ""
$ !
$ ! Acquire the intended username and save it in a temp file. Use it
$ ! to name the log file, and pass it as the first line of input to
$ ! LOGIN.
$ !
$ READ/PROMPT="Username: " SYS$COMMAND USERNAME
$ PID = F$GETJPI (0, "PID")
$ OPEN/WRITE OUTPUT USERNAME'PID'.TMP
$ WRITE OUTPUT USERNAME
$ CLOSE OUTPUT
$ DEFINE/USER SYS$INPUT USERNAME'PID'.TMP
$ SET HOST 0 /LOG='USERNAME'.LOG
$ DELETE USERNAME'PID'.TMP;0
$ LOGOUT
```

You must set up each account for which session auditing is to be enforced as follows:

```
UAF> MODIFY username /FLAGS=CAPTIVE /NOLOCAL /NODIALUP -
     /LGICMD=SYS$SYSROOT:[USER_AUDIT]CHECKAUDIT
```

Because the captive login command procedure assures that the login is coming from the USER_AUDIT account via a SET HOST command, the session is audited.

You may also want to disable batch and network access for each user account to allow only local logins from the USER_AUDIT account, as follows:

```
UAF> MODIFY username/FLAGS=CAPTIVE/NOLOCAL/NODIALUP/NOBATCH -
     /NONETWORK/LGICMD=SYS$SYSROOT:[USER_AUDIT]CHECKAUDIT
```

The CHECKAUDIT.COM procedure verifies that the user is logging into the USER_AUDIT account, as follows:

```
$ ! CHECKAUDIT.COM - ensure that the account is being logged into
$ ! with the session audit account.
$ !
$ IF F$MODE () .NES. "INTERACTIVE" THEN EXIT
$ !
$ ! Verify that the connection originated from the local node and
$ ! from the USER_AUDIT account.
$ !
$ IF F$LOGICAL ("SYS$NODE") .EQS. F$LOGICAL ("SYS$REM_NODE") -
  .AND. F$LOGICAL ("SYS$REM_ID") .EQS. "USER_AUDIT" -
  THEN GOTO OK
$ WRITE SYS$OUTPUT "You may only log into this account with ",-
  "the USER_AUDIT account."
$ LOGOUT
$ !
$ ! When the login has been verified, enable control-Y to
$ ! release the account, invoke the user's LOGIN.COM, and turn
$ ! control over to the user.
$ !
$OK:
$ SET CONTROL_Y
$ IF F$SEARCH ("LOGIN.COM") .EQS. "" THEN EXIT
$ @LOGIN
```

5.9.7 Other Audit Data

In addition to logging security alarms, VMS provides additional data useful in tracking system activity. The system accounting log contains records of all system job terminations, including all interactive, batch, and network jobs, as well as print jobs and other process terminations. Optionally, activations of all or selected images can also be recorded in the accounting log. Further information on the use of the accounting log is available in the *Guide to Setting Up a VMS System* and in the *VMS Accounting Utility Manual.*

Most network operations, such as mail delivery and access to files from remote network nodes, initiate a network server job for which a log file is created. This log file is normally named NETSERVER.LOG and is located in the default directory of the account under which the job ran. You may be able to use the contents of NETSERVER.LOG when tracking down events initiated over the network.

5.10 Ongoing Tasks

Maintaining a secure system requires continuous surveillance. Following are ongoing tasks important to the system manager:

- Use the MONITOR IO report to develop a familiarity with the normal amounts of I/O on your system at various times. Watch for abnormal changes.
- Keep informed of the images installed on your system. Use the VMS Install Utility (INSTALL) to look for unexpected additions.
- Use the AUTHORIZE command SHOW on a regular basis to check for unauthoized user names.
- Use the AUTHORIZE command SHOW/PROXY regularly to quickly recognize all proxy access that you have authorized. Watch for unexpected additions. Remove any remote users who no longer require access. Institute regular communications with system managers at remote nodes.
- Apply the VMS Accounting Utility on a regular basis to give you a basis of normal amounts of processing time. Watch for unexplained changes.
- Regularly check the accounting report produced by ACCOUNTING for known user names, unknown user names, and appropriate hours of system use.
- Develop sufficient familiarity with your system's workload so that you notice normal (as well as abnormal) processing activity occurring at unusual hours.
- Monitor device allocations routinely with the DCL command SHOW DEVICE so that you will immediately notice any that are unexpected.
- Become familiar with the recurring types of batch jobs that run on the batch queues and what times they are most likely to run.
- Monitor the protection and ownership of critical files with the DIRECTORY/SECURITY command. Watch for unexplained changes in each.
- Maintain familiarity with the rights database. Keep current listings so that you can recognize identifiers that have been added or new holders of the current identifiers.
- Remove identifiers that are not in use. Keep the rights database current.
- Regularly review the templates that you may be using to set up UAF records. Make any changes necessary.
- Implement security alarms regularly.
- Try to break into your user's accounts with some obvious password choices.
- When you allow new users to change their initial passwords, check back to see if you can log in with the password you originally assigned. Where necessary, follow up with the user to find out why the change did not occur as requested.

- Try searching unprotected user files for passwords embedded in network access control strings. The password will precede the 3-character terminator ("::). Also search for the noun "password", and see if any passwords are revealed nearby.
- Check that your users are logging out properly. Make physical checks at the end of normal business hours.
- Check that your users have appropriate default protections in place.
- Keep informed about your inventory of magnetic tapes, disks, and program listings. Routinely check that inventory for possible indications that physical security has degraded.
- Keep your office and all important listings locked up.

6 When Your System's Security Has Been Breached

After establishing appropriate security measures for your site, it is still vital to monitor your system for possible security breaches. Following are the most common forms of system attacks:

- Hunting for access lines
- Hunting for passwords
- Attempting a break-in
- Changing or creating UAF records
- Granting/stealing extra privileges
- Introducing apparently innocent software (Trojan Horse software) that is intended to steal user passwords or do other damage to the system.
- Introducing worms in command procedures and programs to gain access to privileged accounts
- Scavenging disks
- Using a node as a gateway to other nodes

This chapter describes how you can recognize when an attack on the system is in progress or has taken place and what countermeasures you can take.

6.1 Indications of Trouble

When your system is vulnerable and possibly under attack, your first indications may come from the following sources:

- Reports from users
- Monitoring the system
- Ongoing auditing applications

The following sections outline problem indicators.

6.1.1 Reports from Users

User observations frequently point to system security problems. A user may contact you with the following situations:

- Files are missing.
- There are unexplained forms of last login messages, such as successful logins the user did not perform or unexplained login failures.
- A user cannot log in, suggesting the user password might have been changed since the last successful login, or some other form of tampering has occurred.

- Break-in evasion appears to be in effect, and the user cannot log in.
- Reports from the SHOW USERS command indicate that the user is logged in on another terminal when the user did not do so.
- A disconnected job message appears during a login for a process the user never initiated.
- Software exists in the user's directories that the user did not write.
- Unexplained changes have been found in the protection or ownership of user files.
- Listings appear that are generated under the user name without the user requesting the listing.
- A sudden reduction occurs in the availability of resources, such as dialup lines.

Follow up promptly when one of these items is reported to you. You must confirm or deny that the condition exists. If you find the complaint is valid, seek a cause and solution.

6.1.2 Monitoring the System

Section 5.10 lists those tasks that can help you detect potential security breaches on your system. The following list details possible warning signs you may uncover while performing the recommended tasks:

- A user appears on the SHOW USERS report that you know could not be currently logged in.
- You observe an unexplained change in the system load.
- You discover media or program listings are missing or notice other indications that physical security has degraded.
- Your locked file cabinet has been tampered with, and the list of authorized users has disappeared.
- You find unfamiliar software in the system executable image library [SYSEXE] or in [SYSLIB].
- You observe unfamiliar images running when you examine the MONITOR SYSTEM report.
- You observe unauthorized user names with SHOW USER. When you examine the listing that AUTHORIZE produces with the SHOW command, you find that those users have been given system access.
- You discover PROXY users that you never authorized.
- The accounting report reveals unusual amounts of processing time expended recently, suggesting outside access.
- You observe unexplained batch jobs on the batch queues.
- You observe unexpected device allocations with the SHOW DEVICE command.
- You observe normal processing activity at unusual hours.

- The UIC-based protection or ACLs change on critical files. Identifiers are added, or holders of identifiers are added to the rights database.
- There is high personnel turnover, low morale, or a problem with attitudes.

All these conditions warrant further investigation. Some indicate that you already have a problem, some may have simple explanations, while others may indicate serious potential problems.

6.2 Routine System Surveillance

VMS provides a number of mechanisms that allow systematic surveillance of the activity in your system. Proper use of such mechanisms should help alert you to problems and allow you to intervene. This section describes the most important system surveillance mechanisms.

6.2.1 Accounting Log

Accounting logs generated by the VMS Accounting Utility can provide early indications of problems. Check your logs for the following:

- Unfamiliar user names
- Unfamiliar patterns of use, such as unusual activity for a particular time of day or day of week
- Use of an unusual amount of resources
- Unfamiliar sources of login, such as network nodes or terminals

6.2.2 Security Auditing

Use the DCL command SET AUDIT to enable alarms. Because security auditing affects system performance, enable security alarms only for the most important events. The following security alarm features are presented in order of decreasing priority and increasing system cost:

1. Enable security auditing for LOGFAIL and BREAKIN. This is the best way to detect probing by outsiders (and insiders looking for accounts). All sites needing security should enable alarms for these events.
2. Enable security auditing for LOGIN. Auditing successful logins from the more suspicious sources like REMOTE and DIALUP provides the best way to track which accounts are being used. An audit record is written before users logging in on a privileged account can disguise their identity.
3. Enable the FILE=FAILURE security audit. This technique audits all file protection violations and is an excellent method of catching probers.
4. Apply ACL-based file access auditing to detect WRITE access to critical system files. The most important files to audit are shown in Table 6–1. Section 4.8.2 presents an example of how to establish security alarm ACEs. You may want to audit only successful access to these files to detect penetrations, or you may want to audit access failures to detect probing as well.

Note that some of the files in Table 6–1 are written during normal system operation. For example, SYSUAF.DAT is written during each login, and SYSMGR.DIR is written when the system boots.

5 Audit use of privilege to access files (either WRITE or all forms of access). Implement the security audit with FILE=(SYSPRV,BYPASS,READALL,GRPPRV). Note that this class of auditing can produce a large volume of output because privileges are often used in normal system operation for such tasks as mail delivery and operator backups.

Table 6–1 System Files Benefiting from ACL-Based File Access Auditing

Device and Directory	File Name
SYS$SYSTEM	AUTHORIZE.EXE
	F11BXQP.EXE
	LOGINOUT.EXE
	DCL.EXE
	JOBCTL.EXE
	JBCSYSQUE.DAT
	SYSUAF.DAT
	NETPROXY.DAT
	RIGHTSLIST.DAT
	STARTUP.COM
SYS$LIBRARY	SECURESHR.EXE
SYS$MANAGER	SYSTARTUP.COM
	VMSIMAGES.DAT
SYS$SYSROOT	[000000]SYSEXE.DIR
	[000000]SYSLIB.DIR
	[000000]SYS$LDR.DIR
	[000000]SYSMGR.DIR

6.3 Handling a Security Breach

There are four phases that security managers experience while handling a security breach, whether the breach actually occurred or was attempted:

1 Detection of a problem

2 Identification of the perpetrator

3 Prevention of further security violations

4 Repair of damage

The following sections describe these phases for both attempted and successful break-ins.

6.3.1 Unsuccessful Break-In Attempts

Unsuccessful break-in attempts include situations where someone has attempted to guess passwords or browse through files.

6.3.1.1 Detection of the Unsuccessful Break-In Attempt

You usually detect break-in attempts through the following sources:

- Reports from users about unexplained login failures
- Unusual system activity or unavailability of dialup lines
- Security alarms for login failures, break-in detection, and file protection violations
- Examination of the break-in database

6.3.1.2 Identifying the Perpetrator

Enabling file auditing simplifies identification of file browsers. If, however, browsing is being initiated from another node in the network, you must inspect the File Access Listener (FAL) logs that correspond to the times of the protection violations. Coordinate your investigation with the security manager at the remote node.

Identifying a perpetrator who is guessing passwords is considerably more difficult, especially when the source is anonymous, as from a dialup line. Usually, you must trade identification for prevention. Often the only way to positively identify an outsider attempting to enter the system requires that you permit further attempts while establishing the perpetrator's identity.

6.3.1.3 Prevention of Break-In Attempts

The prevention phase for this kind of attack involves preventing the would-be intruder from actually gaining access to the system and making future attempts more difficult.

Password Guessing

To reduce the opportunities for successful password guessing, take the following steps:

- Make certain your users choose appropriate passwords. Warn your users that someone is attempting entry. Consider use of the password generator (see Section 5.2.6.5).
- Enable system passwords at the points of entry. While a minor inconvenience to your users, system passwords are the best protection against further probing. If you already had a system password enabled, change it (see Section 5.2.6.2).
- Enable auditing of successful logins to catch the event if the intruder succeeds in getting in (see Section 6.2.2).

File Browsing

To reduce the opportunities for successful file browsing, take the following steps:

- If you can identify the perpetrator, take action as established at your site.
- Warn your users about the importance of adequate protection of their files, and consider inspecting the protection of user files.
- If file browsing from other nodes in the network becomes a persistent problem, eliminate the default FAL account and authorize individual users through proxy login accounts (see Section 7.6).

6.3.1.4 Repair After an Unsuccessful Break-In

No repair of files or data structures is necessary in this class of break-in. Typically, you have lost information to the browser. The value of the information determines the extent of the loss.

6.3.2 Successful Break-In Attempts

A successful security breach can include a successful password guessing scheme, theft or modification of information or system resources, and placement of damaging software on the system. A successful break-in may require a considerable amount of time to repair, depending upon the skill and intent of the perpetrator.

6.3.2.1 Identification of Break-In Perpetrator

Identification is often the most difficult part of handling a break-in. First, you must establish whether the perpetrator is an authorized user or not. This determines the nature of the preventative measures that you will take. However, the distinction between insiders and outsiders may be difficult to achieve.

Tradeoff Between Identification and Prevention

You may have to make a tradeoff between a positive identification of the intruder and preventing future attacks. Often, the data available initially does not allow complete identification. If it is important to identify the perpetrator, you will often find it necessary to permit continued break-ins while you analyze the break-in activity. Step up your auditing. Consider planting traps in system procedures that are under your control (such as SYLOGIN.COM) to obtain additional information. Increase your system backup efforts to permit easier recovery if files become damaged.

Identification of Outsiders

Identifying external break-in perpetrators is particularly difficult, especially if they use any switched forms of communication (such as dialup lines or public data networks). DECnet–VAX provides many features to help you trace the activity through the network back to the source node. If a local terminal is involved, physical surveillance may be appropriate.

When a switched connection is involved, one of the major computer security problems is the telephone system itself. Tracing a telephone or public data network connection is time-consuming. Chasing an intruder through the telephone system is likely to take months and will require the assistance of law enforcement authorities. The advent of independent long-distance

telephone services compounds the problem by increasing the number of organizations you must deal with.

As a result, identifying an outside intruder is usually worthwhile only when you have sustained substantial financial damage. In many cases, it may be more useful if you concentrate on preventing future recurrences of the problem.

6.3.2.2 Prevention of Break-In Attempts

The steps you must take to secure your system after a break-in depend on the nature and source of that break-in. This section describes these steps in order of priority.

- Secure your authorization files. Restore SYSUAF.DAT, NETPROXY.DAT, and RIGHTSLIST.DAT (if damaged) from backups. Alternatively, generate listings of the files and inspect them closely, looking for improper entries, additional privileges, and changed UICs. If you are unsure of when the UAF might have first been modified, inspect it carefully regardless of whether you are using a backup copy or proceeding with the existing one.
- Change passwords. The perpetrator may have discovered passwords by browsing through files or from other nodes in the network and may be using seldom-accessed accounts for personal use. At a minimum, change passwords on all privileged accounts, and have your users appear in person to learn their new passwords. Do not use the same new password for all accounts.
- Clean up your system software. A sophisticated penetrator may have planted ways to provide future access to the system even though you have taken the obvious steps of securing your system. Therefore, you may have to restore selected components of the VMS software from backups or from your VMS distribution kit. If the intruder was an outsider, the only critical component is LOGINOUT.EXE, which validates all entries to the system.

  However, if the intruder was an authorized user, do a complete backup of all system files. Authorized users can make use of a wide variety of "trap doors" in the executive (SYS.EXE), the file system (F11BXQP.EXE), DCL, and other system files. The penetrator may have planted damaging software in any piece of software or command procedure likely to be used by a privileged user. Thus, complete assurance of a "clean" system requires a rather wholesale restoration of files from backups. An alternate strategy is to restore trustworthy copies of the obvious targets of attack and rely on increased auditing for a period of time to catch suspicious events.
- Tighten security. Consider implementing additional security features, such as system passwords, password generation, increased auditing, and more stringent file protection to prevent a recurrence.

6.3.2.3 Repair After a Break-In

Restore corrupted files. Decide whether it is appropriate to do a wholesale restoration of your system's data, or repair problems as they are discovered. Look for modifications to file protection that would have created worm holes and for Trojan horses that were introduced into the system and may still reside there.

7 Security for a DECnet Node

Security in a networking environment is even more sensitive than security in a single system environment. It is also harder to achieve because of operational complexities and the decentralization of control that commonly exist in networks. The larger the network, the more difficult the problem of establishing control and communication between security managers of the numerous nodes.

This chapter provides direction on how security managers can improve network security. Secure nodes and overall network security are even more important to system security than individual node operations and must be monitored and updated regularly.

There are limitations in the degree of security any networking site can expect to achieve due to limitations currently present in networking technology. Being sensitive to potential problems can help you avoid operations that could increase the security exposure in your network. This chapter will help you recognize these problem areas and adjust your operations accordingly.

This chapter assumes the reader is familiar with the information in the *VMS Networking Manual.*

7.1 The Reference Monitor in a Network

Chapter 2 introduces the reference monitor concept. This concept also applies to security in a network of interconnected computer systems. This section first extends the reference monitor concept to the network environment, then summarizes the special considerations that apply in a network, and finally makes the connection between the abstract components of the reference monitor concept and the real elements of a DECnet–VAX network.

In a network, there is a subject on one computer, an object on another, and a network reference monitor that grants the subject access to the object, refers to an authorization database, and develops the required audit trail. Figure 7–1 shows this simplified view of secure access in a network environment.

While, for the most part, the network security mechanisms that DIGITAL employs conform to the abstract model depicted in Figure 7–1, there are some differences. Consider a subject on one node in the network that attempts to access an object on a second node. Since each computer must have its own implementation of the reference monitor model, there will be a phantom object on the system with the real subject (the source machine) and a corresponding phantom subject on the system with the real object (the target machine). The resulting configuration resembles Figure 7–2.

7.1 The Reference Monitor in a Network

Figure 7–1 Simple Diagram of Reference Monitor in a Network

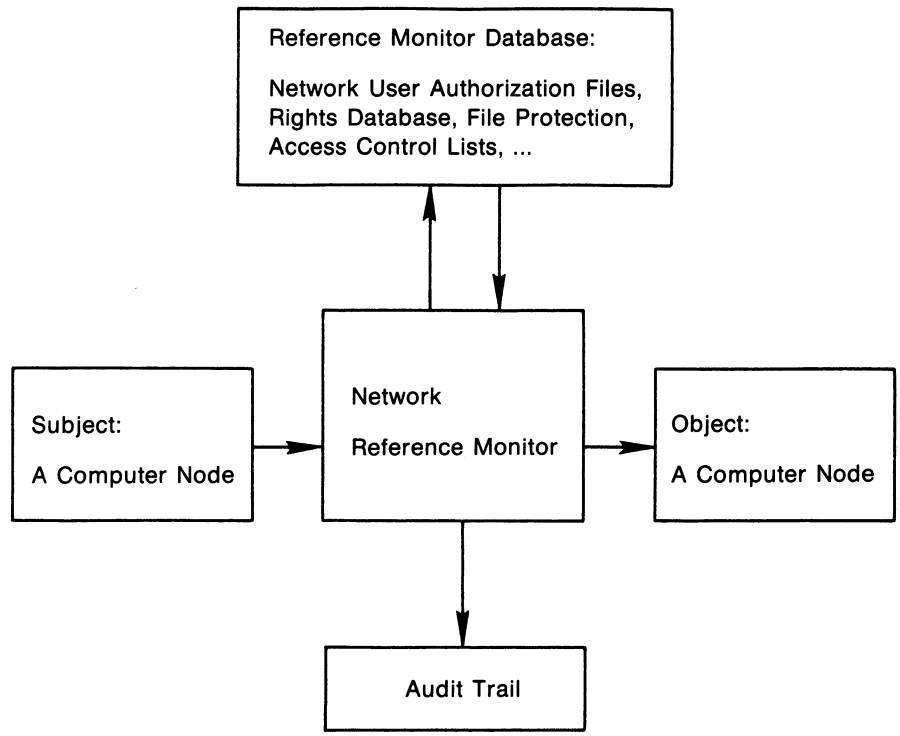
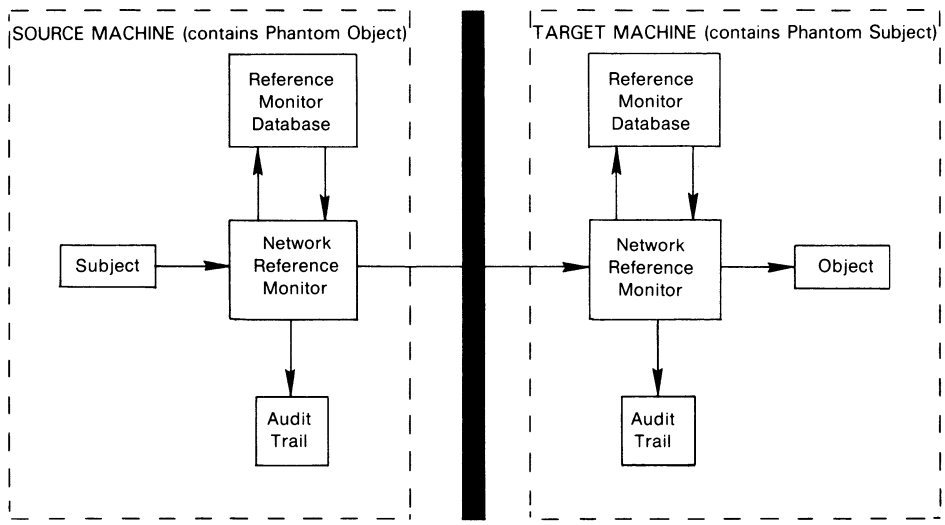


Figure 7–2 Advanced Diagram of the Reference Monitor in a Network



There are three critical requirements for achieving security in a network environment:

- There must be a correspondence between the real subject on the source machine and the phantom subject on the target machine. This correspondence must be managed by the two reference monitors and must be consistent with the security policy intended on the target machine (which is ultimately responsible for protecting the object).
- The authorization database on the target machine must express an access authorization for a phantom subject that corresponds to the real subject on the source machine.
- There must be a protected means of communication between the two reference monitors (source and target) so that correspondence between real and phantom subjects can be reliably established and authenticated.

VMS provides mechanisms to help meet the first two requirements. Mechanisms for meeting the third requirement are discussed in Section 7.1.3.

7.1.1 Establishing Subject Correspondence

VMS and DECnet–VAX provide several mechanisms for establishing a correspondence between a subject or process on a source node and another on a target node. Essentially, the default account mechanisms allow any subject on any node to be placed in correspondence with a default subject on a target node known as the default DECnet account. This subject can in turn gain access to objects on behalf of a requesting subject and return the required information. Because any subject can be placed in correspondence with a default subject on a target node, there is little selectivity or control in the establishment of the correspondence.

Another alternative is the use of explicit or user name/password access control when establishing a subject at the target node. This mechanism restricts access to those objects accessible to the named user, but also causes users' passwords to move about the network without effective protection.

Finally, VMS offers proxy accounts as a means of establishing the correspondence between the real subject on the source node and the phantom subject on the target node. Section 3.2.2 describes proxy accounts. The proxy option requires the target reference monitor to maintain a table of source subjects (by user name and node name) and the corresponding local (target) user names. Then each request from a subject on a source node will be mapped into the creation of a subject representing the corresponding target user. This mechanism offers the explicit control associated with user name/password control, but more adequately protects the passwords.

7.1.2 Specifying Authorizations

The approach used to specify authorization for access to objects depends somewhat on the mechanism for establishing correspondence between subjects. The default account mechanisms essentially create anonymous subjects on the target node. As a result, objects that are to be made accessible to a default account must permit the WORLD user category full access, which leaves the object unprotected.

If either explicit access control or proxy access is used to establish correspondence between subjects, the authorization can be granted to the target subject selected by the user name or proxy. In this case, the full range of VMS authorization mechanisms can be used.

7.1.3 Protecting Communications

It should be clear that the security of network operations depends mostly on the ability of source and target reference monitor mechanisms to communicate in a private, authenticated way. An intruder must not be allowed to observe passwords or to masquerade as a source node that has been granted proxy access.

Protected communication is outside the domain of VMS security. Users can achieve this protection through physical protection of the communication lines or by protecting these lines using encryption. Mechanisms for physical protection (conduit or fiber optics) and encryption are available from third party vendors.

While essential in high-security environments, network security measures such as conduits and encryption are useful at all sites. Many communication media, including most local area networks like Ethernet, are so unprotected that an intruder or authorized user with a personal computer can easily read or forge any information passing over the network.

7.1.4 Summary of VMS Network Security and the Reference Monitor

To summarize, VMS provides, especially through the proxy mechanism, a vehicle for extending user authentication and authorization over a network in a secure, natural, and consistent manner. However, from a system point of view, the network security mechanisms are no better than the protection of the underlying communications. This can be critical in relatively open networks that process sensitive information.

7.2 DECnet–VAX Accounts

DECnet–VAX accounts permit certain types of access to your system from remote nodes without requiring them to specify account and password information. Instead, this information is specified in the DECnet–VAX executor and object databases. Like all accounts, these are controlled through the system authorization file using techniques similar to those used for captive user accounts.

Consider the following general guidelines when you set up accounts for DECnet–VAX use. Detailed examples are given in Examples 7–2 and 7–3.

- DECnet–VAX currently has no requirement for a privileged default account. Do not provide one. Only create a default account for objects when required, rather than setting one up for the executor. DECnet–VAX requires a privileged account for the local use of the Network Management Listener (NML), but you do not need to set it up as a default. It can be specified in a SET EXECUTOR command when required.
- UICs of the network nonprivileged accounts should be unique for each group and user. Furthermore, the group code must exceed the system UIC group number to avoid granting the user the SYSTEM user category for file access. (Ensure this by using group codes greater than the SYSGEN parameter MAXSYSGROUP.)
- Keep the privileges for DECnet–VAX accounts to a minimum. Typically, this means you would only give TMPMBX and NETMBX to nonprivileged accounts.
- Maintain the secrecy of passwords for DECnet–VAX accounts; they need not be known to users of your node or other nodes. Once the password is defined in the authorization file and the DECnet–VAX databases, there will be no need to specify the password.
- Set up the DECnet–VAX accounts with the following AUTHORIZE qualifiers: /FLAGS=(DISCTLY,CAPTIVE,LOCKPWD), /NOINTERACTIVE, and /NOBATCH.
- The account for the File Access Listener (FAL) object should have a group code in its UIC that differs from every other account in the system, including accounts for other DECnet objects. Note that if you have a task object defined, you can prevent outsiders from running arbitrary command procedures on your system if you make the UIC group of the task object different from the UIC group of the default FAL account.

- The member number of the owner UIC of the default directory for the FAL account should be different from the member number of the owner UIC of the FAL account. This ensures that READ and WRITE access is permitted, but CONTROL access is not. Without CONTROL access, a FAL account user cannot change the protection of the directory.
- For any account that does not need to support remote file access, place the following command in the account's login file to log out the account if remote file access is attempted:

  ```
  $ FAL$COMMAND == LOGOUT
  ```

  However, this technique only works when the FAL object is defined to execute the command procedure SYS$SYSTEM:FAL.COM. Since the DECnet–VAX default defines the FAL object to call FAL.EXE directly instead, you will probably need to execute the following DECnet–VAX command:

  ```
  NCP> DEFINE OBJECT FAL FILE FAL.COM
  ```

  This technique can be used with user accounts as well as with the DECnet–VAX accounts to prohibit remote file access by logging out any user who attempts it. Note that this technique shuts off only the remote file access, but allows access to other objects. This is preferable to specifying /NONETWORK in the user's UAF, since that would deny all DECnet use.

7.3 The DECnet–VAX Database

The DECnet–VAX node and circuit databases control how other computers are allowed to connect to your computer. Since a computer connection permits automated assaults on both your own security and that of any other computer in the network, it requires strict control.

To promote the security of the databases, observe the following guidelines:

- Define receive and transmit passwords for all nodes in the database. The receive password defined for a node needs to be known by the manager of that node only if that node can be adjacent. Wherever possible, the transmit and receive passwords should be different and not obvious. (Some operating systems do not permit this.)
- Verification must always be enabled on any circuit that goes outside a locked computer room or to a machine with a different security environment. This is necessary to prevent the node adjacent to you from intercepting mail or circumventing a connection check for the originating node by pretending to be another node in the network. This is particularly important when proxy logins are permitted.
- Do not define default access rights in the database for external nodes. A possible exception to this would be for a server or computer that is a dedicated front end for another computer.
- In general, backup synchronous dialup should not be enabled for autoanswer. Systems that have incoming dialup for production purposes should control which nodes can connect.

7.4 Foreign Network Regulations

Use of the network is restricted in many foreign countries, either by law or by the contract with the major communications division of many governments known as the PTT. Always conform to these regulations. For example, several countries have laws to protect personal data from misuse, including restrictions on moving personal data across country boundaries. This would include moving or remotely accessing personnel databases and might also be interpreted to include such tasks as forwarding job applications. Germany has a law thats forbids transmitting data for processing outside the country.

Similarly, some European laws forbid anyone but the PTT from providing data transmission as a service to customers. In Germany, it is illegal to route data between the X.25 network and the leased line network.

7.5 Specifying DECnet Object Accounts

The following section describes parts of a Network Control Program (NCP) command file that defines some of the usual DECnet objects in the network object database, the UAF entries for the FAL accounts, and the associated login command files. Note that the account used for the FAL object is different from the others. With explicit accounts for all required objects, there is no need for executor default accounts.

Example 7–1 illustrates how the definition for the DECNET and FAL account might appear in the object database.

Example 7–1 Definitions in the Network Object Database

```
!
!  For object FAL account is special
 DEFINE    OBJECT   FAL -
   NUMBER 17-
   FILE SYS$SYSTEM:FAL-
   USER FAL PASSWORD ABCXYZ
!
!  Allow network information
 DEFINE OBJECT NML -
   NUMBER 19 -
   FILE SYS$SYSTEM:NML -
   USER DECNET PASSWORD XYZABC
!
!  Allow MAIL
 DEFINE OBJECT MAIL -
   NUMBER 27 -
   FILE SYS$SYSTEM:MAIL -
   USER DECNET PASSWORD XYZABC
!
!  Allow PHONE
 DEFINE OBJECT PHONE -
   NUMBER 29 -
   FILE SYS$SYSTEM:PHONE -
   USER DECNET PASSWORD XYZABC
!
```

Examples 7–2 and 7–3 present typical entries in SYSUAF.DAT for these two accounts.

Example 7–2 UAF Record for FAL Account

```
Username: FAL                              Owner:  OFFICE
Account:  NETNODE                          UIC:    [301,303] ([DECNET3,FAL])
CLI:      DCL                              Tables:
Default:  DOCD$:[FAL]
LGICMD:   SYS$SYSTEM:FALLOG.COM
Login Flags:  Disctly Defcli Lockpwd Captive
Primary days:   Mon Tue Wed Thu Fri Sat Sun
Secondary days:
Primary   000000000011111111112222  Secondary 000000000011111111112222
Day Hours 012345678901234567890123  Day Hours 012345678901234567890123
Network:  ##### Full access ######            ##### Full access ######
Batch:    -----  No access  ------            -----  No access  ------
Local:    -----  No access  ------            -----  No access  ------
Dialup:   -----  No access  ------            -----  No access  ------
Remote:   -----  No access  ------            -----  No access  ------
Expiration:            (none)    Pwdminimum:  6   Login Fails:     0
Pwdlifetime:           (none)    Pwdchange:  24-JAN-1988 17:19
Last Login:            (none) (interactive), 12-MAR-1988 16:49 (non-interactive)
Maxjobs:         0  Fillm:        16  Bytlm:        12480
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:           0
Maxdetach:       0  BIOlm:        12  JTquota:       1024
Prclm:           0  DIOlm:         6  WSdef:          180
Prio:            4  ASTlm:        16  WSquo:          200
Queprio:         0  TQElm:        10  WSextent:         0
CPU:        (none)  Enqlm:        20  Pgflquo:      25600
Authorized Privileges:
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX
```

Example 7–3 UAF Record for DECNET Account

```
Username: DECNET                           Owner:  DECNET
Account:  NETNODE                          UIC:    [300,300] ([DECNET2,DECNET])
CLI:      DCL                              Tables:
Default:  DOCD$:[DECNET]
LGICMD:   LOGIN.COM
Login Flags:  Disctly Defcli Lockpwd Captive
Primary days:   Mon Tue Wed Thu Fri Sat Sun
Secondary days:
Primary   000000000011111111112222  Secondary 000000000011111111112222
Day Hours 012345678901234567890123  Day Hours 012345678901234567890123
Network:  ##### Full access ######            ##### Full access ######
Batch:    -----  No access  ------            -----  No access  ------
Local:    -----  No access  ------            -----  No access  ------
Dialup:   -----  No access  ------            -----  No access  ------
Remote:   -----  No access  ------            -----  No access  ------
Expiration:            (none)    Pwdminimum:  6   Login Fails:     0
Pwdlifetime:           (none)    Pwdchange:  24-JAN-1988 17:21
Last Login:            (none) (interactive), 23-MAR-1988 16:49 (non-interactive)
Maxjobs:         0  Fillm:        16  Bytlm:        12480
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:           0
Maxdetach:       0  BIOlm:        12  JTquota:       1024
Prclm:           0  DIOlm:         6  WSdef:          180
```

Example 7–3 Cont'd. on next page

Example 7–3 (Cont.) UAF Record for DECNET Account

```
Prio:            4  ASTlm:        16  WSquo:          200
Queprio:         0  TQElm:        10  WSextent:         0
CPU:        (none)  Enqlm:        20  Pgflquo:      25600
Authorized Privileges:
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX
```

Creating and maintaining a FAL account is useful at small sites where security requirements are low to moderate. Sites with high security requirements, or sites where it is difficult to recognize all the intended users, should not use a FAL account. To control which users gain access, these sites might establish one or more proxy accounts for specific purposes. The following section describes how to set up proxy accounts to permit a special type of network login known as proxy login.

7.6 Proxy Logins

As described in Section 3.2.2, you can authorize proxy access when you encounter situations where users on different nodes or in different groups want to share files on your system, and you are reluctant to give out passwords or to set the directory and file protection to WORLD:RWE. With proxy logins, there is no need for passwords to be embedded in commands to copy a file across the network. Also, there is no need for a file's protection code to be set to allow the WORLD category of users READ access to transfer a file. The user enters the following form of the DCL command COPY:

```
$ COPY remotenode::file-spec file-spec
```

You can authorize a remote user access to a default proxy account and up to fifteen other proxy accounts. To copy a file over the network using proxy access from an account other than the default, the user includes the name of the proxy account in the access control string of the DCL command, as follows:

```
$ COPY remotenode"proxyacct"::file-spec file-spec
```

7.6.1 Setting Up Proxy Logins

Two utilities are used to set up proxy logins: AUTHORIZE and NCP. You might want to create a command procedure to assist you in implementing proxy access.

For example, the command procedure could provide the following functions:

- Check if proxy access is enabled for your system
- Create a proxy account for sharing files
- Grant a user access to a proxy account
- Remove a user from access to a proxy account
- Change the default proxy account for a user
- List users authorized to access a proxy account

7.6.1.1 Using the VMS Authorize Utility

To set up proxy logins without using a command procedure, use AUTHORIZE to create or modify the network proxy authorization file, NETPROXY.DAT, that contains the names of all remote users allowed proxy access to the system and the names of all proxy accounts defined for the remote users. Note that the remote user can be specified either by user name or, for non-VMS systems that implement DECnet Phase IV, by UIC. The following commands are used to establish, modify, or display the proxy authorization file:

- CREATE/PROXY
- ADD/PROXY node::remoteuser localuser[,...]
- LIST/PROXY
- MODIFY/PROXY node::remoteuser
- SHOW/PROXY node::remoteuser
- SHOW/PROXY *
- REMOVE/PROXY node::remoteuser

7.6.1.2 Proxy Account Example

When you want to set up a proxy account on your node for use by one or more users at other nodes, you must perform the following steps:

- Decide on the purpose of the account. Decide the name of the local account and which foreign users will be admitted.
- If the local account does not exist, create it with AUTHORIZE; if the account does exist, examine it to ensure it is adequately restricted. Proxy accounts should be restricted so that they prohibit interactive users and batch jobs, which means they should permit only network logins.
- Review the privileges on the account. Generally avoid granting privileges to proxy login accounts. This practice provides a shield between systems in a network in the event one node is penetrated. The fact that proxy logins only provide admittance to nonprivileged accounts at other nodes may help contain the extent of damage if a penetration occurs on one system in the network.
- If the network proxy authorization file NETPROXY.DAT does not exist, create it with the AUTHORIZE command CREATE/PROXY.
- Allow as many remote users as necessary access to the proxy account with the AUTHORIZE command ADD/PROXY. (Exercise caution when authorizing users. Ideally, you should receive a formal request from the security manager at the remote site.)
- Check the default protection on the directory, and customize it as necessary.
- Examine any command procedure used at login time and specified by /LGICMD. Make certain that it follows the recommendations in Section 5.8 for login command procedures in captive accounts. It should reside in a well-protected directory owned by a user other than the owner of the proxy account. It should prohibit WRITE access for those who use the account.

- Notify the security manager at the remote node which users from that node have been authorized for access to your node.

In Example 7–4, the security manager at the node WALNUT wants to create a general access account called GENACCESS. At the same time the manager wants to take steps to allow proxy logins by three users from the node BIRCH: KMAHOGANY, PSUMAC, and WPINE, as well as two users from the node WILLOW: RDOGWOOD and WCHERRY. Assume no network proxy authorization file currently exists.

AUTHORIZE performs certain automatic maintenance functions on the NETPROXY.DAT proxy authorization file. Whenever the user name changes through a RENAME or COPY command, the associated change is made in NETPROXY. Similarly, when you remove an account from SYSUAF.DAT, all entries for which there is a matching local user name are removed from NETPROXY.DAT.

7.6.1.3 Using the VMS Network Control Program (NCP) Utility

Use NCP to control the overall use of proxy login with respect to the executor node and network objects. You can restrict the use of proxy logins on your system by specifying the NCP executor parameters INCOMING PROXY and OUTGOING PROXY, as shown in Table 7–1.

Example 7–4 Example of a Proxy Account

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD GENACCESS /PASSWORD=WHYNADGUM/UIC=[236,043] -
_UAF> /DEVICE=STAFFDEV/DIRECTORY=[GENACCESS] -
_UAF> /OWNER="Security Mgmt"/ACCOUNT=SEC -
_UAF> /FLAGS=(DISWELCOME,DISNEWMAIL,GENPWD,DISMAIL) -
_UAF> /NOBATCH/NOINTERACTIVE/MAXDETACH=8 -
_UAF> /LGICMD=LOGIN/MAXACCTJOBS=10
user record successfully added
identifier for value:[000236,000043] added to RIGHTSLIST.DAT
UAF> CREATE/PROXY
UAF> ADD/PROXY BIRCH::KMAHOGANY  GENACCESS/DEFAULT
record successfully added to NETPROXY.DAT
UAF> ADD/PROXY BIRCH::PSUMAC     GENACCESS/DEFAULT
record successfully added to NETPROXY.DAT
UAF> ADD/PROXY BIRCH::WPINE      GENACCESS/DEFAULT
record successfully added to NETPROXY.DAT
UAF> ADD/PROXY WILLOW::RDOGWOOD   GENACCESS/DEFAULT
record successfully added to NETPROXY.DAT
UAF> ADD/PROXY WILLOW::WCHERRY    GENACCESS/DEFAULT
record successfully added to NETPROXY.DAT
UAF> SHOW/PROXY *::*
 Default proxies are flagged with an *

BIRCH::KMAHOGANY
     GENACCESS *

BIRCH   ::PSUMAC
     GENACCESS *

BIRCH   ::WPINE
     GENACCESS *

WILLOW  ::RDOGWOOD
     GENACCESS *

WILLOW  ::WCHERRY *
     GENACCESS

UAF>  EXIT
{messages}
$ DIRECTORY/SECURITY SYS$STAFF:[000000]GENACCESS.DIR
   .
   .
   .
$ DIRECTORY/SECURITY SYS$STAFF:[GENACCESS]LOGIN.COM
   .
   .
   .
```

Table 7–1 Executor Proxy Parameter Values

Parameter	Meaning
INCOMING PROXY enabled	Allows proxy login access from the remote node to the local node
INCOMING PROXY disabled	Prevents proxy login access from the remote node to the local node
OUTGOING PROXY enabled	Allows the local system to initiate proxy login access to the remote system
OUTGOING PROXY disabled	Prevents the local system from initiating proxy login access to the remote system

By default, both incoming and outgoing proxy login access are enabled at the local system.

You can also control proxy login access by network objects by setting the value of the object parameter PROXY in the OBJECT database. Specify proxy login access for a particular network object (such as MAIL or FAL) only when the desired proxy access is different from that defined in the EXECUTOR database. Refer to the *VMS Networking Manual* for information on using NCP to modify the executor and object databases.

The control parameters are found in the executor and object databases. They each are part of the CHARACTERISTICS display that you can generate with the following command:

```
$ RUN SYS$SYSTEM:NCP SHOW CHAR OBJ FAL
```

The EXECUTOR database contains the INCOMING PROXY and OUTGOING PROXY parameters. These parameters are used to supply values for other parameters when they are not explicitly set up for a given node or object. These parameters make it easy to set up the DECnet–VAX configuration database.

Proxy access will not function for nodes that have privileged or nonprivileged access control specified (parameters NONPRIVILEGED (or PRIVILEGED) USER, PASSWORD, and ACCOUNT). The concept of outbound proxy access conflicts with the concept of default outbound access control strings. This conflict occurs on the destination node. When a connect message containing non-null access control strings is received, the receiving node has no way of knowing whether those strings were specified explicitly by the user or were defaults provided by the source-node operating system; when access control strings are passed in the connect message, they are used, and proxy access is inhibited.

The USER, PASSWORD, and ACCOUNT parameters should rarely be used. They are still needed if default access is to be provided to nodes that cannot provide default inbound access control. VMS nodes are all capable of providing default inbound access control (in addition to proxy access) by setting the NONPRIV USER, PASSWORD, and ACCOUNT parameters in the EXECUTOR database.

If outbound proxy access is implicitly set for a node to OUTGOING PROXY ENABLED in the EXECUTOR database, the USER, PASSWORD, and ACCOUNT parameters may still be set up for that node. In this case, outbound proxy access to that node will be inhibited since the DECnet–VAX connect message will contain non-null access control.

The OBJECT database contains the PROXY parameter to control proxy access to and from individual objects in the network. The value for this parameter is taken from the EXECUTOR INCOMING PROXY and EXECUTOR OUTGOING PROXY parameters if it has not been given an explicit value or if a given object is not defined in the database.

7.6.1.4 Conditions for Proxy Access

For proxy access to be allowed, five conditions must be satisfied. If any of these conditions are not met, the default DECnet account is used.

- The EXECUTOR DEFAULT PROXY parameter for the initiating node must be either BOTH or OUTGOING.
- The OBJECT PROXY parameter for the initiating node must be either BOTH or OUTGOING.
- The EXECUTOR DEFAULT PROXY parameter for the destination node must be either BOTH or INCOMING.
- The OBJECT PROXY parameter for the destination node must be either BOTH or INCOMING.
- There must be an entry in NETPROXY.DAT on the destination node for the initiating node-user pair. For example, if the account HYDRA on the destination node of CRAB will permit proxy access for user CLAW on node LOBSTER, the listing of NETPROXY.DAT for node CRAB would include the following entry:

```
LOBSTER::CLAW
      HYDRA *
```

7.6.2 Special Proxy Access Considerations

Proxy access is a selective merging of the authorization databases of the affected systems. Therefore, the security is only as good as the security of the least secure node involved.

Although proxy access eliminates passwords going over the network, it is possible for a personal computer to bypass the proxy login mechanism by impersonating one of the authorized nodes. For this reason, the parameter INCOMING for proxy should not be used on vital nodes. Never set up a proxy account with privileges that could damage your system. (Proxy accounts should be unprivileged.) In general, timesharing nodes should not permit proxy access from standalone nodes.

7.7 Sharing Files in the Network Environment

The easiest way for a user to transfer a text file to another user is to invoke the VMS Mail Utility and send the user a copy of the file. This method is reasonably secure, since passwords need not be revealed, and the original protection of the file is not changed. The receiving user simply includes a new file name with the MAIL command EXTRACT/NOHEADER to place a copy in the user's own directory. The copy automatically acquires the user's default protection. The user would then use the MAIL command DELETE to remove the copy from the mail file.

This procedure is inappropriate for nontext files, such as binary files. Alternate procedures become more useful once greater numbers of files and users become involved.

Sites should discourage users from sharing passwords and changing file and directory protection codes to grant the WORLD category READ or EXECUTE access. Grant BYPASS or READALL privileges cautiously. The only secure method for sharing and exchanging files in the network environment is to set up proxy accounts and place ACLs on the directories and files.

7.7.1 Multiple Remote Users Seek Access for a Single Task

A network manager may need to admit a number of users from outside nodes into a directory on the local node for a specific task. In this situation, the security manager creates a proxy account and adds the proxy access to admit the outsiders into that one account. There may also be a number of users on the local node who need to share the files in this account's directory. To provide that access while protecting the files from outsiders, place ACLs on the directory and files.

Consider an example where a central depository is needed for sales update information that a number of users scattered throughout the corporation need to read. The security manager at the node (BERTHA) where the files will reside, creates the special account SALES_READER.SALES_READER is set up as a captive account with mail disabled. The default directory is [SALESINFO], which has the following default protection code:

```
(S:RWED,O:RWED,G:R,W)
```

Note that this protection code permits users in the same group as SALES_READER on the home node BERTHA to read the files. Furthermore, only the users in the system category, the owner category, or those who have privileges that give them such access, can update the files in the directory. ACLs are used to further define the access, as shown later.

Next, the security manager uses the AUTHORIZE command ADD/PROXY to add the proxy access for the outside users. For example, to extend proxy access to user JACKSON on node DEXTER and user GOODWIN on node BANGOR, the commands would be as follows:

```
ADD/PROXY DEXTER::JACKSON   SALES_READER/DEFAULT
ADD/PROXY BANGOR::GOODWIN   SALES_READER/DEFAULT
```

If later it becomes clear that other users at the home node BERTHA need access and they do not belong to the same group as SALES_READER, ACLs could be added to the files in the directory [SALESINFO]. For example, suppose R_GRANT needs CONTROL access to all the files and J_MAGOON

needs READ access to all the files. The following two DCL commands would define the ACL for the directory and then propagate it to all existing files:

```
$ SET ACL/ACL=-
_$ ((IDENTIFIER=R_GRANT,OPTIONS=DEFAULT,ACCESS=CONTROL),-
_$ (IDENTIFIER=J_MAGOON,OPTIONS=DEFAULT,ACCESS=READ))-
_$ [000000]SALESINFO.DIR
$ SET FILE/ACL/DEFAULT *.*;*
```

7.7.2 Remote Users from One Node Require Single Account Access

When all (or nearly all) users at a remote node require access to one of your accounts, specify proxy access to the account with the following form of the AUTHORIZE command ADD/PROXY:

```
ADD/PROXY  remote-node::*  local-account/DEFAULT
```

Check to be certain that there are no guest accounts or other undesired accounts at the remote node. If you discover there are a few exceptions at the remote node, you cannot simply remove the extra users with REMOVE/PROXY commands because the preceding ADD/PROXY command creates a single entry in NETPROXY.DAT. Use the following technique to exclude specific individual users at the remote node from access to the proxy account:

```
ADD/PROXY  remote-node::*  local-account/DEFAULT
ADD/PROXY  remote-node::FRED  DECNET/DEFAULT
ADD/PROXY  remote-node::GEORGE  DECNET/DEFAULT
```

In the preceding example, all users on the remote node use the specified proxy account except users FRED and GEORGE who are directed to use the default DECNET account.

7.7.3 A Few Outside Users Require Access for Multiple Purposes

The preceding techniques work well when there are several outside users requiring access for one purpose. When a small number of outside users need access for multiple purposes involving files needing special protection, set up access to multiple proxy accounts and apply extensive ACLs.

For example, a large corporation with many branch offices might find it desirable to establish several proxy accounts for specific purposes for sharing. Assume the central corporation offices want to grant two key users from their two East Coast nodes READ and WRITE access to the project files for code name LEVIGRAY and READ only access to the BETSEYHARLOW project files. At the same time, there are three users from the West Coast who need READ access to those LEVIGRAY files and require READ and WRITE access to the BETSEYHARLOW files. Only two users from the central office will have full access rights to the LEVIGRAY files and two other users from headquarters will have full access rights to the BETSEYHARLOW files. For working purposes, the situation could be represented in tabular form as shown in Example 7–5.

Example 7–5 Example of Protected File Sharing in a Network

Access Requirements to CENTRL::PROJ:[DESGN_PROJECTS]
Owned by [DESIGNERS,MGR]

Users & Nodes	Subdirectory LEVI Project Files LEVIGRAY*.*	Subdirectory BETSEY Project Files BETSEYHARLOW*.*
FRISCO::ALBION	R	RW
FRISCO::ELTON	R	RW
LA::IRVING	R	RW
CENTRL::DIANTHA	RWED	NONE
CENTRL::BRITTANIA	RWED	NONE
CENTRL::ALBERT	NONE	RWED
CENTRL::DELIA	NONE	RWED
BOS::AYLMER	RW	R
WASH::LAVINA	RW	R

The following solution uses five proxy accounts in addition to the four local accounts on CENTRL, plus ACLs on the directory, subdirectories, and files. First, the security manager at headquarters uses AUTHORIZE to create new proxy accounts on node CENTRL, for the remote users ALBION, ELTON, IRVING, AYLMER, and LAVINA. These accounts should be captive, disallow mail, and be restricted to network access only. The accounts are even restricted to a subset of DCL through command language tables. The default directory should be [DESGN_PROJECTS] for each user. The manager decides it makes sense to put them into the DESIGNER group to match their proposed uses of the files.

Presumably, accounts already exist for DIANTHA, BRITTANIA, ALBERT, and DELIA. They need not necessarily belong to the same group. They will be informed which device and directory to use for their work.

The next step is to add the proxy records to the network proxy authorization file with the following AUTHORIZE commands:

```
ADD/PROXY FRISCO::ALBION ALBION/DEFAULT
ADD/PROXY FRISCO::ELTON ELTON/DEFAULT
ADD/PROXY LA::IRVING IRVING/DEFAULT
ADD/PROXY BOS::AYLMER AYLMER/DEFAULT
ADD/PROXY WASH::LAVINA LAVINIA/DEFAULT
```

The security manager at node CENTRL places an ACL on the top-level directory for [DESGN_PROJECTS] with the following DCL command:

```
$ SET ACL/ACL=(DEFAULT_PROTECTION,S:RWED,O,G,W) -
_$ [000000]DESGN_PROJECTS.DIR
```

This ensures that no one outside of the SYSTEM category of users can gain any UIC-based access to the files in the directory or any of the subdirectories unless they possess the BYPASS privilege. In fact, this restriction applies to those five users in the group DESIGNERS as well. The plan is for all files to possess ACLs that will admit the select group of users. It is desirable to propagate this protection code to all the files in this directory and its subdirectories. (The ACLs that will be placed on the files for further protection will take precedence when one of these users actually seeks access to a file.)

Two subdirectories are created in [DESGN_PROJECTS]:

- [DESGN_PROJECTS.LEVI]
- [DESGN_PROJECTS.BETSEY]

Next, the security manager uses the VMS ACL Editor to place the following additional ACEs in the ACL for the top-level directory:

```
DESGN_PROJECTS.DIR

(IDENTIFIER=DIANTHA,OPTIONS=PROTECTED,ACCESS=EXECUTE)
(IDENTIFIER=BRITTANIA,OPTIONS=PROTECTED,ACCESS=EXECUTE)
(IDENTIFIER=ALBERT,OPTIONS=PROTECTED,ACCESS=EXECUTE)
(IDENTIFIER=DELIA,OPTIONS=PROTECTED,ACCESS=EXECUTE)
(IDENTIFIER=AYLMER,OPTIONS=PROTECTED,ACESS=EXECUTE)
(IDENTIFIER=LAVINA,OPTIONS=PROTECTED,ACCESS=EXECUTE)
(IDENTIFIER=ALBION,OPTIONS=PROTECTED,ACCESS=EXECUTE)
(IDENTIFIER=ELTON,OPTIONS=PROTECTED,ACCESS=EXECUTE)
(IDENTIFIER=IRVING,OPTIONS=PROTECTED,ACCESS=EXECUTE)
```

These protected ACEs ensure that only the select nine users can access the top-level directory. Since no one receives WRITE or DELETE access to the top directory through the ACL, the directory and subdirectories are generally protected from deletion and renaming of files. (Of course, the SYSTEM category of user obtains WRITE and DELETE access through the UIC-based protection.)

Next, the security manager must create ACLs on the subdirectories. The ACEs that are required are shown for their respective subdirectories:

```
[DESGN_PROJECTS]LEVI.DIR

(IDENTIFIER=DIANTHA,OPTIONS=PROTECTED,ACCESS=READ+WRITE+EXECUTE+CONTROL)
(IDENTIFIER=DIANTHA,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=BRITTANIA,OPTIONS=PROTECTED,ACCESS=READ+WRITE+EXECUTE+CONTROL)
(IDENTIFIER=BRITTANIA,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=AYLMER,OPTIONS=PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=AYLMER,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=LAVINA,OPTIONS=PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=LAVINA,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=ALBION,OPTIONS=PROTECTED,ACCESS=READ)
(IDENTIFIER=ALBION,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ)
(IDENTIFIER=ELTON,OPTIONS=PROTECTED,ACCESS=READ)
(IDENTIFIER=ELTON,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ)
(IDENTIFIER=IRVING,OPTIONS=PROTECTED,ACCESS=READ)
(IDENTIFIER=IRVING,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ)

[DESGN_PROJECTS]BETSEY.DIR

(IDENTIFIER=ALBERT,OPTIONS=PROTECTED,ACCESS=READ+WRITE+EXECUTE+CONTROL)
(IDENTIFIER=ALBERT,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=DELIA,OPTIONS=PROTECTED,ACCESS=READ+WRITE+EXECUTE+CONTROL)
(IDENTIFIER=DELIA,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=ALBION,OPTIONS=PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=ALBION,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=ELTON,OPTIONS=PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=ELTON,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=IRVING,OPTIONS=PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=IRVING,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ+WRITE)
(IDENTIFIER=AYLMER,OPTIONS=PROTECTED,ACCESS=READ)
(IDENTIFIER=AYLMER,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ)
(IDENTIFIER=LAVINA,OPTIONS=PROTECTED,ACCESS=READ)
(IDENTIFIER=LAVINA,OPTIONS=DEFAULT+PROTECTED,ACCESS=READ)
```

You will note that both preceding ACLs include two ACEs for each identifier. The first ACE controls the access to the subdirectory. It denies delete access for the protection of the subdirectory and is not propagated to all the files created in the subdirectory. The second ACE for each identifier will automatically propagate to all files added to their respective subdirectories because of the inclusion of the OPTIONS=DEFAULT option specification. Furthermore, the option PROTECTED ensures that all the ACEs are protected from deletion except by specific action. At this point, all the groundwork has been completed. Over time, files are added to the subdirectories. Thus, when the user LAVINA in Washington enters the following DCL command, the file LEVIGRAYMEM3.MEM is printed at node WASH:

```
$ COPY CENTRL::LEVIGRAYMEM3.MEM LP:
```

However, any attempts by user LAVINA to edit the file BETSEYHARLOWMEM8.MEM would fail, because user LAVINA is denied WRITE access through the ACL.

If there were many users involved in this scheme, it would soon become worthwhile to grant additional identifiers to the users. For example, each user who would be allowed READ access to the files in the LEVI subdirectory might be given the identifier LEVI_READER, and so forth. The ACLs could then be shortened.

8 Security Concerns on a Cluster

This chapter describes security concerns to security managers on clustered VMS systems. You should be familiar with the information in the *VMS VAXcluster Manual.*

The *VMS VAXcluster Manual* describes the person or team of persons who manage a VAXcluster as the cluster manager. The cluster manager is a specialized system manager. The security manager for a VAXcluster generally requires the training and skills of the cluster manager. At some VAXcluster sites, the security manager role is also performed by the cluster manager. At other sites, there may be one or more security managers in addition to a cluster management team.

When a site separates the security manager function, coordination, cooperation, and communication are vital. As in previous chapters, this chapter uses the title of security manager to refer to individuals who have the responsibility for system security, regardless of what other responsibilities they may or may not hold.

8.1 Overview of Clusters and Security Considerations

Clustered VMS systems refer to those systems using VMS hardware and software that permits sharing of disks, resources, and a common operating system among various VAX computers. The computers are said to be joined in a VAXcluster. There are two types of VAXclusters: homogeneous and nonhomogeneous (or heterogeneous). A *homogeneous* VAXcluster refers to one in which the operating system environment is identical on each member node. On a *nonhomogeneous* VAXcluster, each node has a unique environment.

The fact that a node is part of a VAXcluster generally has little significance. Because each node in a VAXcluster appears to operate as a single system, all security features previously described apply equally well to any node in the cluster. When a security manager implements the feature on one node of a homogeneous cluster, all nodes are affected. Each cluster node mediates access by its subjects to all objects in the cluster. In effect, the cluster operates within a single security perimeter, with the reference monitor on each node acting as a gateway through that perimeter.

There is, however, one area where the actions the cluster manager takes in setting up the VAXcluster can affect the security operations of the system. This concerns the creation and management of the authorization database. This chapter describes those security implications and provides recommendations.

8.2 Authorization Database Considerations

On a VAXcluster, there are advantages when all elements of the user authorization data exist in a common database. These authorization elements include the system and network user authorization files and the rights database, which are present on all VMS systems, and the optional autologin file, SYSALF.DAT, which some sites create with the procedure SYS$MANAGER:ALFMAINT.COM. (Section 5.2.8 describes how to use ALFMAINT.COM.) If you create an autologin file, consider maintaining it in a common authorization database with your authorization files and rights database. On a clustered system, the autologin file must include the cluster node name as a prefix to the terminal name. For example, the terminal TTA0 on node WILLOW would be represented as WILLOW$TTA0.

The reasons for maintaining your authorization elements on a common disk are as follows:

- Centralized management of the data is facilitated, which saves time and errors.
- A common system disk allows you to maintain consistent, up-to-date system software for all nodes in the cluster. That is, if you want any of the three primary elements (NETPROXY, SYSUAF, or RIGHTSLIST) to be common for all nodes, you must have only one copy of each file in the cluster. This requirement reflects the fact that AUTHORIZE performs some automatic maintenance functions on the data.

While you are not required to set up a common shared disk for these files, you are encouraged to do so. If you do not, remember that coordination is required in your choices for UICs, group numbers, and identifiers. They all must be unique. Each user should have the same UIC, group number, and set of identifiers defined on every node.

8.3 Building a Common User Environment

Refer to the *VMS VAXcluster Manual* for guidelines for building a common user environment. The procedures depend on the initial state of your system.

8.4 File Sharing Considerations

When disks are shared, the file system works locally on each node to perform file protection checking. By setting up separate authorization files for each node, you could overlook the actual user privileges and access. A shared disk is no more protected than it is at its least protected node. If you maintain separate authorization files on each node in the cluster, ensure that user privileges are common across all copies of the UAF.

8.5 Using DECnet Between Cluster Nodes

While VAXclusters offer special communication facilities for the most common operations (file sharing and lock management), other VMS features may require the use of DECnet to be used across a cluster. For example, you might need to access disks that are not cluster-accessible or use higher-level features available through DCL commands such as SHOW USERS.

For maximum consistency in DECnet operations, set up a proxy database that maps users into their own accounts when they initiate DECnet operations. Thus, for each node in a homogeneous cluster, you would add a proxy file entry using the following AUTHORIZE command:

```
ADD/PROXY node::* */DEFAULT
```

If you are running a nonhomogeneous cluster, you will need a more complex arrangement of proxies to cross-map users as they are authorized.

8.6 Summary

In summary, security operations are enhanced on a VAXcluster when all authorization data resides on a common shared disk. The files of concern are:

- SYSUAF.DAT
- NETPROXY.DAT
- RIGHTSLIST.DAT
- SYSALF.DAT (if it exists)

Each user must have the same UIC, group number and set of identifiers defined on each cluster node.

On a shared disk, the protection of a file from a specific user cannot effectively exceed the maximum access that user can gain from one of the nodes.

In all respects, VMS security features operate the same on clustered systems as they do on nonclustered systems.

A Privileges

A.1 User Privileges

This appendix describes all privileges available on VMS systems, including the abilities passed by each privilege and the users who should receive them.

A.1.1 ACNT Privilege

Only a user who has the ACNT privilege can create subprocesses or detached processes in which accounting is disabled. Thus, only such a privileged user can enter the DCL command RUN with the /NOACCOUNTING qualifier or inhibit accounting in the Create Process ($CREPRC) system service.

A.1.2 ALLSPOOL Privilege

ALLSPOOL allows the user's process to allocate a spooled device by executing the Allocate Device ($ALLOC) system service or by using the DCL command ALLOCATE.

The $ALLOC system service lets a process allocate, or reserve, a device for its exclusive use. A shareable mounted device cannot be allocated.

Grant this privilege only to users who need to perform logical or physical I/O operations to a spooled device. Ordinarily, the privilege of allocating a spooled device is granted only to symbionts.

A.1.3 ALTPRI Privilege

ALTPRI allows the user's process to:

- Increase its own base priority
- Set the base priority of another process to a value higher than that of the target process

The base priority is increased by executing the Set Priority ($SETPRI) system service or the DCL command SET PROCESS/PRIORITY. As a rule, this system service lets a process set its own base priority or the base priority of another process. However, one process can only set the priority of a second process if one of the following conditions applies:

- The process calling the $SETPRI system service has the same UIC as the target process.
- The calling process has process control privilege (GROUP or WORLD) over the target process.

With ALTPRI, a process can create a process with a priority higher than its own. It creates such a process by using an optional argument to the Create Process ($CREPRC) system service or to the DCL command RUN.

Do not grant this privilege widely; if unqualified users have the unrestricted ability to set base priorities, fair and orderly scheduling of processes for execution can easily be disrupted.

A.1.4 BUGCHK Privilege

The use of this privilege should be restricted to DIGITAL-supplied system software that uses the VMS Bugcheck Facility. This privilege allows the process to make bugcheck error log entries.

A.1.5 BYPASS Privilege

BYPASS allows the user's process read, write, execute, and delete access to all files, bypassing UIC-based and ACL protection.

Grant this privilege with extreme caution, as it overrides all file protection. It should be reserved for use by either well-tested, reliable programs and command procedures or the system backup operation (see the *Guide to Maintaining a VMS System* for a discussion of backup operations). SYSPRV (see below) is adequate for interactive use, as it ultimately grants access to all files while still providing access checks.

A.1.6 CMEXEC Privilege

CMEXEC allows the user's process to execute the Change Mode to Executive ($CMEXEC) system service.

This system service lets a process change its access mode to executive, execute a specified routine, and then return to the access mode that was in effect before the system service was called. While in executive mode, the process is allowed to execute the Change Mode to Kernel ($CMKRNL) system service.

Grant this privilege only to users who need to gain access to protected and sensitive data structures and internal functions of the operating system. If unqualified users have unrestricted access to sensitive data structures and functions, the operating system and service to other users can be easily disrupted. Such disruptions can include failure of the system, destruction of the database, and exposure of confidential information.

A.1.7 CMKRNL Privilege

CMKRNL allows the user's process to execute the Change Mode to Kernel ($CMKRNL) system service.

This system service lets a process change its access mode to kernel, execute a specified routine, and then return to the access mode that was in effect before the system service was called.

Grant this privilege only to users who need to execute privileged instructions or who need to gain access to the most protected and sensitive data structures and functions of the operating system. If unqualified users have unrestricted use of privileged instructions and unrestricted access to sensitive data structures and functions, the operating system and service to other users can be easily disrupted. Such disruptions can include failure of the system, destruction of the database, and exposure of confidential information.

A.1.8 DETACH Privilege

Users can create detached processes that have their own UIC without this privilege, provided the users do not exceed their MAXJOBS and MAXDETACH quotas. However, the DETACH privilege becomes valuable when a user wants to specify a different UIC for the detached process. There is no restriction on the UIC that can be specified for a detached process if you have the DETACH privilege. Thus, there are no restrictions on the files and directories to which a detached process can gain access.

DETACH allows the user's process to create detached processes by executing the Create Process ($CREPRC) system service. Detached processes remain in existence even after the user who created them has logged off the system.

An example of a detached process is the process created by the system for a user when the user logs in to the system.

A.1.9 DIAGNOSE Privilege

DIAGNOSE allows the user to run online diagnostic programs and to intercept and copy all messages written to the error log file.

A.1.10 EXQUOTA Privilege

EXQUOTA allows the space taken by the user's files on given disk volumes to exceed any usage quotas set for the user (as determined by UIC) on those volumes.

A.1.11 GROUP Privilege

GROUP privilege allows the user's process to affect other processes in its own group by executing the following process control system services: Suspend Process ($SUSPND), Resume Process ($RESUME), Delete Process ($DELPRC), Set Priority ($SETPRI), Wake ($WAKE), Schedule Wakeup ($SCHDWK), Cancel Wakeup ($CANWAK), and Force Exit ($FORCEX). The user's process is also allowed to examine other processes in its own group by executing the Get Job/Process Information ($GETJPI) system service. The user with the GROUP privilege can issue the following DCL commands for other processes in its group: SET QUEUE, DELETE/ENTRY, STOP/ENTRY, and SET PROCESS.

GROUP privilege is not needed for a process to exercise control over, or to examine, subprocesses that it created or other detached processes of its UIC. You should, however, grant this privilege to users who need to exercise control over each other's processes and operations.

A.1.12 GRPNAM Privilege

GRPNAM allows the user's process to insert names into the logical name table of the group to which the process belongs and to delete names from that table by the use of the following logical name system services: Create Logical Name ($CRELNM) and Delete Logical Name ($DELLNM).

In addition, the privileged user can use the DCL commands ASSIGN and DEFINE to add names to the group logical name table, the DCL command DEASSIGN to delete names from the table, and the /GROUP qualifier of the DCL command MOUNT to share volumes among group members.

Do not grant this privilege to all users of the system because it allows the user to create an unlimited number of group logical names. When unqualified users have the unrestricted ability to create group logical names, excessive use of system dynamic memory can degrade system performance. In addition, a user with the GRPNAM privilege can interfere with the activities of other users in the same group by creating definitions of commonly used logical names such as SYS$SYSTEM.

A.1.13 GRPPRV Privilege

GRPPRV allows a process access to a file using the file's SYSTEM protection field when the process's group matches the group of the file owner. GRPPRV also allows a process to change the protection of any file whose owner group matches the process's group. This privilege also allows a process to change the ownership of objects within the process's group.

Grant this privilege only to users who function as group managers. Note that if any member of a group holds any of the privileges in the "all" category, then any other member of that group who holds GRPPRV privilege can gain control of the system by indirectly acquiring that privilege. (The privileges in the "all" category have the potential to control the system and are described in Section 5.3.6.)

A.1.14 LOG_IO Privilege

LOG_IO allows the user's process to execute the Queue I/O Request ($QIO) system service to perform logical-level I/O operations. LOG_IO privilege is also required for certain device control functions, such as setting permanent terminal characteristics.

Usually, user I/O requests are handled indirectly by use of an I/O package such as VAX Record Management Services. However, to increase their control over I/O operations and to improve the efficiency of I/O operations, skilled users sometimes prefer to handle directly the interface between their process and a system I/O driver program. They can do this by executing the Queue I/O Request system service; in many instances, the operation called for is a logical-level I/O operation. Note that logical level functions are permitted without LOG_IO privilege on a device mounted with the /FOREIGN qualifier and on nonfile-structured devices.

Grant this privilege only to users who need it because it allows a process to access data anywhere on the selected volume without the benefit of any file structuring. If this privilege is given to unqualified users who have no need for it, the operating system and service to other users can be easily disrupted. Such disruptions can include the destruction of information on the

system device, the destruction of user data, and the exposure of confidential information.

A.1.15 MOUNT Privilege

MOUNT allows the user's process to execute the mount volume QIO function. The use of this function should be restricted to system software supplied by DIGITAL.

A.1.16 NETMBX Privilege

NETMBX allows the user to perform functions related to a DECnet computer network. Grant this privilege to general users.

A.1.17 OPER Privilege

OPER privilege allows the user to use the Operator Communication Manager (OPCOM) process, as follows: to reply to user's requests, to broadcast messages to all terminals logged in, to designate terminals as operators' terminals and specify the types of messages to be displayed on these operators' terminals, and to initialize and control the log file of operators' messages. In addition, this privilege lets the user set devices spooled and create and control queues.

Grant this privilege only to the operators of the system. These are the users who respond to the requests of ordinary users, who tend to the needs of the system's peripheral devices (mounting reels of tape and changing printer forms), and who attend to all the other day-to-day chores of system operation. (A nonprivileged user can log in on the console terminal to respond to operator requests, for example, to mount a tape.)

A.1.18 PFNMAP Privilege

PFNMAP allows the user's process to map to specific pages of physical memory or I/O device registers, no matter who is using the pages or registers.

Exercise caution in granting this privilege. If unqualified users have unrestricted access to physical memory, the operating system and service to other users can be easily disrupted. Such disruptions can include failure of the system, destruction of the database, and exposure of confidential information.

A.1.19 PHY_IO Privilege

PHY_IO allows the user's process to execute the Queue I/O Request ($QIO) system service to perform physical-level I/O operations.

Usually, users' I/O requests are handled indirectly by use of an I/O package such as VAX Record Management Services. However, to increase their control over I/O operations and to improve the efficiency of their applications, skilled users sometimes prefer to handle directly the interface between their process and a system I/O driver program. They can do this by executing the Queue

I/O Request system service; in many instances, the operation called for is a physical-level I/O operation.

Grant the PHY_IO privilege only to users who need it; this privilege should be granted even more carefully than the LOG_IO privilege. If this privilege is given to unqualified users who have no need for it, the operating system and service to other users can be easily disrupted. Such disruptions can include the destruction of information on the system device, the destruction of user data, and the exposure of confidential information.

A.1.20 PRMCEB Privilege

PRMCEB allows the user's process to create or delete a permanent common event flag cluster by executing the Associate Common Event Flag Cluster ($ASCEFC) or Delete Common Event Flag Cluster ($DLCEFC) system service. Common event flag clusters enable cooperating processes to communicate with each other and thus provide the means of synchronizing their execution.

Do not grant this privilege to all users of the system because it allows the user to create an unlimited number of permanent common event flag clusters. A permanent cluster remains in the system even after the creating process has been terminated and continues to use up a portion of system dynamic memory. When many users have the unrestricted ability to create permanent common event flag clusters, the excessive use of system dynamic memory can degrade system performance.

A.1.21 PRMGBL Privilege

PRMGBL allows the user's process to create permanent global sections by executing the Create and Map Section ($CRMPSC) system service. In addition, the user with this privilege (plus CMKRNL and SYSGBL privileges) can use the VMS Install Utility.

Global sections are shared structures that can be mapped simultaneously in the virtual address space of many processes. All processes see the same code or data. Global sections are used for reentrant subroutines or data buffers.

Grant this privilege with care. If permanent global sections are not explicitly deleted, they tie up space in the global section and global page tables, which are limited resources.

A.1.22 PRMMBX Privilege

PRMMBX allows the user's process to create or delete a permanent mailbox by executing the Create Mailbox and Assign Channel ($CREMBX) system service or the Delete Mailbox ($DELMBX) system service.

Mailboxes are buffers in virtual memory that are treated as if they were record-oriented I/O devices. A mailbox is used for general interprocess communication.

Do not grant PRMMBX granted to all users of the system. Permanent mailboxes are not automatically deleted when the creating processes are deleted and, thus, continue to use a portion of system dynamic memory.

A.1.23 PSWAPM Privilege

PSWAPM allows the user's process to control whether it can be swapped out of the balance set by executing the Set Process Swap Mode ($SETSWM) system service. Not only must a process have this privilege to lock itself in the balance set (that is, to disable swapping), but also to unlock itself (that is, to enable swapping).

With this privilege, a process can create a process that is locked in the balance set (process swap mode disabled) by using an optional argument to the Create Process ($CREPRC) system service or, when the DCL command RUN is used to create a process, by using a qualifier of the RUN command.

Grant this privilege only to users who need to lock a process in memory for performance reasons. Typically, this will be a real-time process. If unqualified users have the unrestricted ability to lock processes in the balance set, physical memory can be held unnecessarily and thereby degrade system performance.

A.1.24 READALL Privilege

READALL allows the process to bypass existing restrictions that would otherwise prevent the process from reading a file. However, unlike the BYPASS privilege, which permits writing and deleting, READALL only permits reading of the file and control operations (such as changing protection and writing the backup date).

Grant this privilege to operators so they can perform system backups. The implications of this privilege are the same as those for the SYSPRV privilege.

A.1.25 SECURITY Privilege

SECURITY privilege allows a process to perform security related functions such as enabling or disabling security audits or setting the system password.

Grant this privilege only to security managers. Irresponsible users who obtain this privilege can subvert the system's security auditing and can lock out users through improper application of system passwords.

A.1.26 SETPRV Privilege

SETPRV allows the user's process to create processes whose privileges are greater than its own by executing the Create Process ($CREPRC) system service with an optional argument, or by issuing the DCL command RUN to create a process. A user with this privilege can also execute the DCL command SET PROCESS/PRIVILEGES to obtain any desired privilege.

Exercise the same caution in granting SETPRV as in granting any other privilege, since SETPRV allows the user to enable any or all privileges.

A.1.27 SHARE Privilege

SHARE privilege allows processes to assign channels to devices allocated to other processes.

Grant this privilege only to system processes such as print symbionts. This privilege would allow an irresponsible user to interfere with the operation of devices belonging to other users.

A.1.28 SHMEM Privilege

SHMEM allows the user's process to create global sections and mailboxes (permanent and temporary) in multiport memory if the process also has appropriate PRMGBL, PRMMBX, SYSGBL, and TMPMBX privileges. Just as in local memory, the space required for a multiport memory temporary mailbox counts against the buffered I/O byte count limit (BYTLM) of the process.

A.1.29 SYSGBL Privilege

SYSGBL allows the user's process to create system global sections by executing the Create and Map Section ($CRMPSC) system service. In addition, the user with this privilege (plus the CMKRNL and PRMGBL privileges) can use the VMS Install Utility.

Exercise caution in granting this privilege. System global sections require space in the global section and global page tables, which are limited resources.

A.1.30 SYSLCK Privilege

SYSLCK allows the user's process to lock systemwide resources with the Enqueue Lock Request ($ENQ) system service. Grant this privilege to users who need to run programs that lock resources in the systemwide resource name space.

Exercise caution in granting this privilege. Users who hold the SYSLCK privilege can interfere with the synchronization of system software and all other user software as well.

A.1.31 SYSNAM Privilege

SYSNAM allows the user's process to insert names into the system logical name table and to delete names from that table by using the Create Logical Name ($CRELNM)and Delete Logical Name ($DELLNM) system services. This privilege also permits the creation of executive mode logical names.

In addition, the user with this privilege can use the DCL commands ASSIGN and DEFINE to add names to the system logical name table, and can use the DEASSIGN command to delete names from the table.

Grant this privilege only to the system operators or to system programmers who need to define system logical names (such as names for user devices, library directories, and the system directory). For example, to mount or dismount a system volume, which entails defining a system logical name, you must have the SYSNAM privilege. Note that a user with SYSNAM privilege could redefine such critical system logical names as SYS$SYSTEM and SYSUAF, thus gaining control of the system.

A.1.32 SYSPRV Privilege

SYSPRV allows the user to access objects by the SYSTEM protection field and to change the owner UIC and protection of a file. Even if a file is protected against system access, the user with the SYSPRV privilege can simply change the file's protection to gain access to it.

Exercise caution in granting this privilege. Normally you would only grant this privilege to system managers and security managers. If unqualified users have system access rights, the operating system and service to others can be easily disrupted. Such disruptions can include failure of the system, destruction of the database, and exposure of confidential information.

A.1.33 TMPMBX Privilege

TMPMBX allows the user's process to create a temporary mailbox by executing the Create Mailbox and Assign Channel ($CREMBX) system service.

Mailboxes are buffers in virtual memory that are treated as if they were record-oriented I/O devices. A mailbox is used for general interprocess communication. Unlike a permanent mailbox, which must be explicitly deleted, a temporary mailbox is deleted automatically when it is no longer referenced by any process.

Grant this privilege to all users of the system to facilitate interprocess communication. System performance is not likely to be degraded by permitting the creation of temporary mailboxes, because their number is controlled by limits on the use of system dynamic memory (BYTLM quota).

A.1.34 VOLPRO Privilege

VOLPRO allows the user to: (1) initialize a previously used volume with an owner UIC different from the user's own UIC; (2) override the expiration date on a tape or disk volume owned by another user; (3) use the /FOREIGN qualifier to mount a Files-11 volume owned by another user; (4) override the owner UIC protection of a volume. The VOLPRO privilege only permits control over volumes the user can mount or initialize. Volumes mounted with the /SYSTEM qualifier are safe from the user with the VOLPRO privilege as long as the user does not also have the SYSNAM privilege.

Exercise extreme caution in granting the VOLPRO privilege. If unqualified users can override volume protection, the operating system and service to others can be disrupted. Such disruptions can include destruction of the database and exposure of confidential information.

A.1.35 WORLD Privilege

WORLD privilege allows the user's process to affect other processes both inside and outside its group by executing the following process control system services: Suspend Process ($SUSPND), Resume Process ($RESUME), Delete Process ($DELPRC), Set Priority ($SETPRI), Wake ($WAKE), Schedule Wakeup ($SCHDWK), Cancel Wakeup ($CANWAK), and Force Exit ($FORCEX). The user's process is also allowed to examine processes outside its own group by executing the Get Job/Process Information ($GETJPI) system service. The user with the WORLD privilege can issue the DCL commands SET QUEUE, DELETE/ENTRY, STOP/ENTRY, and SET PROCESS for all other processes.

To exercise control over subprocesses that it created or to examine these subprocesses, a process needs no special privilege. To affect or to examine other processes inside its own group, a process needs only the GROUP privilege. To affect or examine processes outside its own group, a process needs the WORLD privilege.

B Using the User Data Areas in UAF Records

Users can use VMS Record Management Services (RMS) to access UAF records for the storage and retrieval of up to 255 bytes of user data. The format of a UAF record is defined in the module $UAFDEF in SYS$LIBRARY:LIB.MLB. You may also find it useful to read the UAF$ section of SYS$LIBRARY:LIB.REQ, which contains commented structure definitions.

Access UAF records sequentially through the following keys:

- User name—The primary key is the user name (as specified to AUTHORIZE), a character field of size UAF$S_USERNAME located at relative offset UAF$T_USERNAME.
- UIC—The secondary key is the UIC, located at relative offset UAF$L_UIC, consisting of two binary subfields of one word each. The subfields are named UAF$W_GRP (high-order word) and UAF$W_MEM (low-order word).

To place data in a UAF record, take the following steps, which are designed to protect programs against future changes to the format of the UAF:

1. Read the UAF record.
2. Check the value of UAF$W_USRDATOFF. If it is zero, insert the current size of the record, as found in the VMS RMS record access block (RAB), into UAF$W_USRDATOFF. (In VMS Version 5.0, the system initializes this field to zero. Inserting the current size of the record has the effect of placing the user data at the end of the record. However, future changes to the UAF might require the system to fix the location of the user data. In this event, the system would initialize UAF$W_USRDATOFF to a nonzero value which the user must not change.)
3. Insert the user data at the relative offset pointed to by UAF$W_USRDATOFF. The data must take the form of a counted string, and the first byte must specify the number of bytes that follow. The total number of bytes must not exceed 256.
4. Modify the size of the record in the RAB to reflect the addition of the user data area, unless the current size of the record exceeds the end of the user data area (that is, the contents of UAF$W_USRDATOFF plus the length of the user data). (In Version 5.0, the user data area, when specified as previously outlined, resides at the end of the record, so that modification of the RAB is essential. It is possible, however, that a future version of AUTHORIZE might fix the location of the user data area and its size (at 256 bytes), and locate additional system data after the user data. In this event, the user must not modify the size of the record.)
5. Update the record.

The user can now access the counted string by referring to UAF$W_USRDATOFF. For additional updates, the user must not modify UAF$W_USRDATOFF. However, if the record size changes, the user does modify the first byte of the counted string and the record size in the RAB, bearing in mind the considerations for future changes.

To avoid interfering with normal system operations, open the UAF to allow concurrent update; that is, specify to VMS RMS SHR=(GET,PUT,UPD,DEL). To update a record, you must lock it when you read it.

Note: The format of the UAF record and the way in which the system modifies it is subject to change in future versions of VMS. (For format changes, however, DIGITAL will provide a utility to update old UAFs.) Adherence to the preceding guidelines will minimize reprogramming in the event of UAF record changes.

C Protection for VMS System Files

The following display of protection codes and ownership corresponds to values that DIGITAL supplies for the system files following a normal installation. Monitor these values regularly to ensure that no tampering has occurred. (The DCL commands DIRECTORY/SECURITY/OUTPUT and DIFFERENCES facilitate such checks.) This display was produced from the system manager's account with the following DCL command:

```
$ DIRECTORY/SECURITY/OUTPUT=SYSTEM_FILES.LIS SYS$SYSROOT:[*...]

Directory DB:[SYS0]

SYS$LDR.DIR;1          [SYSTEM]          (RWE,RWE,RE,RE)
SYS$STARTUP.DIR;1      [SYSTEM]          (RWE,RWE,RE,RE)
SYSCBI.DIR;1           [SYSTEM]          (RWE,RWE,RE,RE)
SYSERR.DIR;1           [SYSTEM]          (RWE,RWE,RE,RE)
SYSEXE.DIR;1           [SYSTEM]          (RWE,RWE,RE,RE)
SYSHLP.DIR;1           [SYSTEM]          (RWE,RWE,RE,RE)
SYSLIB.DIR;1           [SYSTEM]          (RWE,RWE,RE,RE)
SYSMAINT.DIR;1         [SYSTEM]          (RWE,RWE,RE,RE)
SYSMGR.DIR;1           [SYSTEM]          (RWE,RWE,RE,RE)
SYSMSG.DIR;1           [SYSTEM]          (RWE,RWE,RE,RE)
SYSTEST.DIR;1          [SYSTEM]          (RWE,RWE,RE,RE)
SYSUPD.DIR;1           [SYSTEM]          (RWE,RWE,RE,RE)

Total of 12 files.

Directory DB:[SYS0.SYS$LDR]

CLUSTRLOA.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
CNDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
CONINTERR.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
CPULOA.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
CRDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
CTDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
CVDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
CWDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DBDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DDDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DLDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DMDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DQDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DRDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DSDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DUDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DVDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DXDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DYDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DZDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
ERRORLOG.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
ESDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
ETDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
EVENT_FLAGS_AND_ASTS.EXE;1
                       [SYSTEM]          (RWED,RWED,RWED,RE)
EXCEPTION.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
EXEC_INIT.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
FBDRIVER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
```

```
FPEMUL.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
FYDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
IMAGE_MANAGEMENT.EXE;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
IO_ROUTINES.EXE;1     [SYSTEM]          (RWED,RWED,RWED,RE)
LADRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
LCDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
LIDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
LOCKING.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
LOGICAL_NAMES.EXE;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
LPDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
LTDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
MBXDRIVER.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
MESSAGE_ROUTINES.EXE;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
NDDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
NETDRIVER.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
NODRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
PADRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
PAGE_MANAGEMENT.EXE;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
PBDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
PDDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
PEDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
PRIMITIVE_IO.EXE;1    [SYSTEM]          (RWED,RWED,RWED,RE)
PROCESS_MANAGEMENT.EXE;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
PUDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
RECOVERY_UNIT_SERVICES.EXE;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
RMS.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
RTTDRIVER.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
RXDRIVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
SCSLOA.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
SECURITY.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
SYS.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
SYSDEVICE.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSGETSYI.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLICENSE.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA410.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA41D.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA41W.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA650.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA65D.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA65W.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA730.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA750.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA780.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA790.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA8NN.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOA8SS.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOAUV1.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOAUV2.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOAWS1.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOAWS2.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSLOAWSD.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSTEM_DEBUG.EXE;1    [SYSTEM]          (RWED,RWED,RWED,RE)
SYSTEM_PRIMITIVES.EXE;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
SYSTEM_SYNCHRONIZATION.EXE;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
SYSTEM_SYNCHRONIZATION_MIN.EXE;1
```

```
                        [SYSTEM]            (RWED,RWED,RWED,RE)
SYSTEM_SYNCHRONIZATION_UNI.EXE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
TFDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
TMDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
TSDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
TTDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
TUDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
TVDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
VAXEMUL.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$SYSTEM_IMAGES.DATA;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
WORKING_SET_MANAGEMENT.EXE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
WPDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
XADRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
XDDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
XEDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
XFDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
XGDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
XIDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
XMDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
XQDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
YCDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
YEDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
YFDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
YIDRIVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)

Total of 105 files.

Directory DB:[SYS0.SYS$STARTUP]

LICENSE_CHECK.EXE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$BASEENVIRON-050_LIB.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$BASEENVIRON-050_SMISERVER.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$BASEENVIRON-050_VMS.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$CONFIG-050_CACHE_SERVER.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$CONFIG-050_CSP.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$CONFIG-050_ERRFMT.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$CONFIG-050_JOBCTL.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$CONFIG-050_LMF.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$CONFIG-050_OPCOM.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$CONFIG-050_VMS.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$INITIAL-050_CONFIGURE.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$INITIAL-050_LIB.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$INITIAL-050_VMS.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$LAYERED.DAT;1       [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$LPBEGIN-050_STARTUP.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$PHASES.DAT;1        [SYSTEM]            (RWED,RWED,RWED,RE)
```

```
VMS$SYSFILES-050_VMS.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMS$VMS.DAT;1           [SYSTEM]            (RWED,RWED,RWED,RE)

Total of 19 files.

Directory DB:[SYS0.SYSERR]

ERRLOG.SYS;1            [SYSTEM]            (RWED,RWED,RE,)
ERRSNAP.COM;1           [SYSTEM]            (RWED,RWED,RWED,RE)

Total of 2 files.

Directory DB:[SYS0.SYSEXE]

ACC.EXE;1               [SYSTEM]            (RWED,RWED,RWED,RE)
ACLEDT.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
AGEN$FEEDBACK.EXE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
ANALIMDMP.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
ANALYZBAD.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
ANALYZOBJ.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
ANALYZRMS.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
AUTHORIZE.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
BACKUP.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
BADBLOCK.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
BOOT58.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
BOOTBLOCK.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
CDU.EXE;1               [SYSTEM]            (RWED,RWED,RWED,RE)
CHECKSUM.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
CIA.EXE;1               [SYSTEM]            (RWED,RWED,RWED,RE)
CONFIGURE.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
CONVERT.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
COPY.EXE;1              [SYSTEM]            (RWED,RWED,RWED,RE)
CREATE.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
CREATEFDL.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
CSP.EXE;1               [SYSTEM]            (RWED,RWED,RWED,RE)
CVTNAFV5.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
DBLMSGMGR.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
DCL.EXE;1               [SYSTEM]            (RWED,RWED,RWED,RE)
DCLDEF.STB;1            [SYSTEM]            (RWED,RWED,RWED,RE)
DELETE.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
DIFF.EXE;1              [SYSTEM]            (RWED,RWED,RWED,RE)
DIRECTORY.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
DISKQUOTA.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
DISMOUNT.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
DSRINDEX.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
DSRTOC.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
DTR.COM;1               [SYSTEM]            (RWED,RWED,RWED,RE)
DTRECV.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
DTSEND.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
DUMP.EXE;1              [SYSTEM]            (RWED,RWED,RWED,RE)
EDF.EXE;1               [SYSTEM]            (RWED,RWED,RWED,RE)
EDT.EXE;1               [SYSTEM]            (RWED,RWED,RWED,RE)
ERF.EXE;1               [SYSTEM]            (RWED,RWED,RWED,RE)
ERFADPTR.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
ERFBRIEF.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
ERFBUS.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
ERFCNTRL.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
ERFDISK.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
ERFDISK2.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
ERFMISC.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
ERFMSCP.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
ERFRLTIM.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
ERFSUMM.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
```

```
ERFTAPE.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
ERFTAPE2.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
ERFUVAX.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
ERFVAX7XX.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
ERFVX8200.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
ERFVX8600.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
ERFVX87XX.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
ERRFMT.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
ERRSNAP.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
EVL.COM;1              [SYSTEM]          (RWED,RWED,RWED,RE)
EVL.EXE;1              [SYSTEM]          (RWED,RWED,RWED,RE)
EXCHANGE.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
F11AACP.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
F11BXQP.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
FAL.COM;1              [SYSTEM]          (RWED,RWED,RWED,RE)
FAL.EXE;1              [SYSTEM]          (RWED,RWED,RWED,RE)
FILESERV.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
HLD.COM;1              [SYSTEM]          (RWED,RWED,RWED,RE)
HLD.EXE;1              [SYSTEM]          (RWED,RWED,RWED,RE)
HSCPAD.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
IMGDEF.STB;1           [SYSTEM]          (RWED,RWED,RWED,RE)
INIT.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
INPSMB.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
INSTALL.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
JBC$UPGRADE.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
JOBCTL.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
LALOAD.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
LALOADER.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
LATCP.EXE;1            [SYSTEM]          (RWED,RWED,RWED,RE)
LATSYM.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
LIBRARIAN.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
LINK.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
LMF.EXE;1              [SYSTEM]          (RWED,RWED,RWED,RE)
LOGINOUT.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
MACRO32.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
MAIL$UPGRADE.EXE;1     [SYSTEM]          (RWED,RWED,RWED,RE)
MAIL.COM;1             [SYSTEM]          (RWED,RWED,RWED,RE)
MAIL.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
MAILEDIT.COM;1         [SYSTEM]          (RWED,RWED,RWED,RE)
MAIL_SERVER.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
MASK_IMAGE.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
MESSAGE.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
MIRROR.COM;1           [SYSTEM]          (RWED,RWED,RWED,RE)
MIRROR.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
MODPARAMS.DAT;1        [SYSTEM]          (RWED,RWED,RWED,RE)
MOM.COM;1              [SYSTEM]          (RWED,RWED,RWED,RE)
MOM.EXE;1              [SYSTEM]          (RWED,RWED,RWED,RE)
MONITOR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
MSCP.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
MTAAACP.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
NCP.EXE;1              [SYSTEM]          (RWED,RWED,RWED,RE)
NCS.EXE;1              [SYSTEM]          (RWED,RWED,RWED,RE)
NCSSHR.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
NETACP.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
NETDEF.STB;1           [SYSTEM]          (RWED,RWED,RWED,RE)
NETSERVER.COM;1        [SYSTEM]          (RWED,RWED,RWED,RE)
NETSERVER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
NICONFIG.COM;1         [SYSTEM]          (RWED,RWED,RWED,RE)
NICONFIG.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
```

```
NML.COM;1                [SYSTEM]            (RWED,RWED,RWED,RE)
NML.EXE;1                [SYSTEM]            (RWED,RWED,RWED,RE)
OPCCRASH.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
OPCOM.EXE;1              [SYSTEM]            (RWED,RWED,RWED,RE)
PATCH.EXE;1              [SYSTEM]            (RWED,RWED,RWED,RE)
PHONE.COM;1              [SYSTEM]            (RWED,RWED,RWED,RE)
PHONE.EXE;1              [SYSTEM]            (RWED,RWED,RWED,RE)
PRTSMB.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
QUEMAN.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
RECLAIM.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
RECOVER.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
REMACP.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
RENAME.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
REPLY.EXE;1              [SYSTEM]            (RWED,RWED,RWED,RE)
REQSYSDEF.STB;1          [SYSTEM]            (RWED,RWED,RWED,RE)
REQUEST.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
RMSDEF.STB;1             [SYSTEM]            (RWED,RWED,RWED,RE)
RMSREC$RU_RECOVER.EXE;1
                         [SYSTEM]            (RWED,RWED,RWED,RE)
RTB.EXE;1                [SYSTEM]            (RWED,RWED,RWED,RE)
RTPAD.EXE;1              [SYSTEM]            (RWED,RWED,RWED,RE)
RUNDET.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
RUNOFF.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
SCSDEF.STB;1             [SYSTEM]            (RWED,RWED,RWED,RE)
SDA.EXE;1                [SYSTEM]            (RWED,RWED,RWED,RE)
SDLNPARSE.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
SEARCH.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
SET.EXE;1                [SYSTEM]            (RWED,RWED,RWED,RE)
SETPO.EXE;1              [SYSTEM]            (RWED,RWED,RWED,RE)
SETRIGHTS.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
SETSHOACL.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
SETWATCH.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
SHOW.EXE;1               [SYSTEM]            (RWED,RWED,RWED,RE)
SHUTDOWN.COM;1           [SYSTEM]            (RWED,RWED,RWED,RE)
SHWCLSTR.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
SMGBLDTRM.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
SMGMAPTRM.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
SMGTERMS.TXT;1           [SYSTEM]            (RWED,RWED,RWED,RE)
SMISERVER.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
SMPUTIL.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
SORTMERGE.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
SRTTRN.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
STABACCOP.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
STABACKUP.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
STACONFIG.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
STANDCONF.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
STARTUP.COM;1            [SYSTEM]            (RWED,RWED,RE,)
STARTUP.INS;1            [SYSTEM]            (RWED,RWED,RWED,RE)
STASYSGEN.EXE;1          [SYSTEM]            (RWED,RWED,RWED,RE)
STOPREM.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
SUBMIT.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
SUCCESS.COM;1            [SYSTEM]            (RWED,RWED,RWED,RE)
SUMSLP.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
SYS.MAP;1                [SYSTEM]            (RWED,RWED,RWED,RE)
SYS.STB;1                [SYSTEM]            (RWED,RWED,RWED,RE)
SYSBOOT.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
SYSDEF.STB;1             [SYSTEM]            (RWED,RWED,RWED,RE)
SYSGEN.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
SYSINIT.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
SYSMAN.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
SYSUAF.TEMPLATE;1        [SYSTEM]            (RWED,RWED,RWED,RE)
TECO32.EXE;1             [SYSTEM]            (RWED,RWED,RWED,RE)
```

```
TERMTABLE.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
TERMTABLE.TXT;1       [SYSTEM]          (RWED,RWED,RWED,RE)
TERTIARY_VMB.EXE;1    [SYSTEM]          (RWED,RWED,RWED,RE)
TFF$MASTER.DAT;1      [SYSTEM]          (RWED,RWED,RWED,RE)
TFU.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
TPU.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
TYPE.EXE;1            [SYSTEM]          (RWED,RWED,RWED,RE)
UNLOCK.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
VERIFY.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
VMB.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
VMBUVAX1P.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
VMOUNT.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
VMSHELP.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
VMSPARAMS.DAT;1       [SYSTEM]          (RWED,RWED,RWED,RE)
VPM.EXE;1             [SYSTEM]          (RWED,RWED,RWED,RE)
WP.EXE;1              [SYSTEM]          (RWED,RWED,RWED,RE)
WRITEBOOT.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
XFLOADER.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)

Total of 187 files.

Directory DB:[SYSO.SYSHLP]

ACLEDT.HLB;1          [SYSTEM]          (RWED,RWED,RWED,RE)
ANLRMSHLP.HLB;1       [SYSTEM]          (RWED,RWED,RWED,RE)
DEBUGHLP.HLB;2        [SYSTEM]          (RWED,RWED,RWED,RE)
DISKQUOTA.HLB;1       [SYSTEM]          (RWED,RWED,RWED,RE)
EDFHLP.HLB;1          [SYSTEM]          (RWED,RWED,RWED,RE)
EDTHELP.HLB;2         [SYSTEM]          (RWED,RWED,RWED,RE)
EDTVT100.DOC;1        [SYSTEM]          (RWED,RWED,RWED,RE)
EDTVT52.DOC;1         [SYSTEM]          (RWED,RWED,RWED,RE)
EVE$HELP.HLB          [SYSTEM]          (RWED,RWED,RWED,RE)
EVE$KEYHELP.HLB       [SYSTEM]          (RWED,RWED,RWED,RE)
EXAMPLES.DIR;1        [SYSTEM]          (RWE,RWE,RE,RE)
EXCHNGHLP.HLB;2       [SYSTEM]          (RWED,RWED,RWED,RE)
HELPLIB.HLB;2         [SYSTEM]          (RWED,RWED,RWED,RE)
INSTALHLP.HLB;1       [SYSTEM]          (RWED,RWED,RWED,RE)
LATCP.HLB;1           [SYSTEM]          (RWED,RWED,RWED,RE)
MAILHELP.HLB;2        [SYSTEM]          (RWED,RWED,RWED,RE)
MNRHELP.HLB;1         [SYSTEM]          (RWED,RWED,RWED,RE)
NCPHELP.HLB;2         [SYSTEM]          (RWED,RWED,RWED,RE)
PATCHHELP.HLB;1       [SYSTEM]          (RWED,RWED,RWED,RE)
PHONEHELP.HLB;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SDA.HLB;1             [SYSTEM]          (RWED,RWED,RWED,RE)
SHWCLHELP.HLB;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SYSGEN.HLB;1          [SYSTEM]          (RWED,RWED,RWED,RE)
SYSMANHELP.HLB;1      [SYSTEM]          (RWED,RWED,RWED,RE)
TECO.HLB;1            [SYSTEM]          (RWED,RWED,RWED,RE)
TFF$TFUHELP.HLB       [SYSTEM]          (RWED,RWED,RWED,RE)
TPUHELP.HLB;1         [SYSTEM]          (RWED,RWED,RWED,RE)
UAFHELP.HLB;1         [SYSTEM]          (RWED,RWED,RWED,RE)
VMSTLRHLP.HLB;1       [SYSTEM]          (RWED,RWED,RWED,RE)

Total of 29 files.

Directory DB:[SYSO.SYSHLP.EXAMPLES]

ADDRIVER.MAR;1        [SYSTEM]          (RWED,RWED,RWED,RE)
ADDUSER.COM;1         [SYSTEM]          (RWED,RWED,RWED,RE)
BACKUSER.COM;1        [SYSTEM]          (RWED,RWED,RWED,RE)
CONNECT.COM;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DB_REQUESTER.C;1      [SYSTEM]          (RWED,RWED,RWED,RE)
DB_REQUESTER.MAR;1    [SYSTEM]          (RWED,RWED,RWED,RE)
DB_SERVER.C;1         [SYSTEM]          (RWED,RWED,RWED,RE)
```

```
DB_SERVER.MAR;1        [SYSTEM]        (RWED,RWED,RWED,RE)
DOD_ERAPAT.MAR;1       [SYSTEM]        (RWED,RWED,RWED,RE)
DRCOPY.PRM;1           [SYSTEM]        (RWED,RWED,RWED,RE)
DRCOPYBLD.COM;1        [SYSTEM]        (RWED,RWED,RWED,RE)
DRMAST.MAR;1           [SYSTEM]        (RWED,RWED,RWED,RE)
DRMASTER.FOR;1         [SYSTEM]        (RWED,RWED,RWED,RE)
DRSLAVE.FOR;1          [SYSTEM]        (RWED,RWED,RWED,RE)
DRSLV.MAR;1            [SYSTEM]        (RWED,RWED,RWED,RE)
DTE_DF03.MAR;1         [SYSTEM]        (RWED,RWED,RWED,RE)
DTE_DF112.MAR;1        [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$ADVANCED.TPU;1     [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$BUILD.TPU;1        [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$CORE.TPU;1         [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$EDIT.TPU;1         [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$EDT.TPU;1          [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$EXTEND.TPU;1       [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$EXTRAS.TPU;1       [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$FILE.TPU;1         [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$FORMAT.TPU;1       [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$HELP.TPU;1         [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$MASTER.FILE;1      [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$OPTIONS.TPU;1      [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$SHOW.TPU;1         [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$VERSION.DAT;1      [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$WILDCARD.TPU;1     [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$WINDOWS.TPU;1      [SYSTEM]        (RWED,RWED,RWED,RE)
EVE$WPS.TPU;1          [SYSTEM]        (RWED,RWED,RWED,RE)
GBLSECUFO.MAR;1        [SYSTEM]        (RWED,RWED,RWED,RE)
LABCHNDEF.FOR;1        [SYSTEM]        (RWED,RWED,RWED,RE)
LABIO.OPT;1            [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOACQ.FOR;1         [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOCIN.MAR;1         [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOCIN.OPT;1         [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOCOM.FOR;1         [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOCOMP.COM;1        [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOCON.FOR;1         [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOLINK.COM;1        [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOPEAK.FOR;1        [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOSAMP.FOR;1        [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOSEC.FOR;1         [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOSTAT.FOR;1        [SYSTEM]        (RWED,RWED,RWED,RE)
LABIOSTRT.COM;1        [SYSTEM]        (RWED,RWED,RWED,RE)
LABMBXDEF.FOR;1        [SYSTEM]        (RWED,RWED,RWED,RE)
LBRDEMO.COM;1          [SYSTEM]        (RWED,RWED,RWED,RE)
LBRDEMO.FOR;1          [SYSTEM]        (RWED,RWED,RWED,RE)
LBRMAC.MAR;1           [SYSTEM]        (RWED,RWED,RWED,RE)
LOGIN.COM              [SYSTEM]        (RWED,RWED,RWED,RE)
LPATEST.FOR;1          [SYSTEM]        (RWED,RWED,RWED,RE)
LPMULT.B32;1           [SYSTEM]        (RWED,RWED,RWED,RE)
MGRMENU.COM;1          [SYSTEM]        (RWED,RWED,RWED,RE)
MONITOR.COM;1          [SYSTEM]        (RWED,RWED,RWED,RE)
MONSUM.COM;1           [SYSTEM]        (RWED,RWED,RWED,RE)
MSCPMOUNT.COM;1        [SYSTEM]        (RWED,RWED,RWED,RE)
PEAK.FOR;1             [SYSTEM]        (RWED,RWED,RWED,RE)
SCRFT.MAR;1            [SYSTEM]        (RWED,RWED,RWED,RE)
SYSGTTSTR.MSG;1        [SYSTEM]        (RWED,RWED,RWED,RE)
TDRIVER.MAR;1          [SYSTEM]        (RWED,RWED,RWED,RE)
TESTLABIO.FOR;1        [SYSTEM]        (RWED,RWED,RWED,RE)
```

```
USSDISP.MAR;1          [SYSTEM]              (RWED,RWED,RWED,RE)
USSLNK.COM;1           [SYSTEM]              (RWED,RWED,RWED,RE)
USSTEST.MAR;1          [SYSTEM]              (RWED,RWED,RWED,RE)
USSTSTLNK.COM;1        [SYSTEM]              (RWED,RWED,RWED,RE)
XADRIVER.MAR;1         [SYSTEM]              (RWED,RWED,RWED,RE)
XALINK.MAR;1           [SYSTEM]              (RWED,RWED,RWED,RE)
XAMESSAGE.MAR;1        [SYSTEM]              (RWED,RWED,RWED,RE)
XATEST.COM;1           [SYSTEM]              (RWED,RWED,RWED,RE)
XATEST.FOR;1           [SYSTEM]              (RWED,RWED,RWED,RE)
XIDRIVER.MAR;1         [SYSTEM]              (RWED,RWED,RWED,RE)

Total of 75 files.

Directory DB:[SYSO.SYSLIB]

ACLEDIT.INI;1          [SYSTEM]              (RWED,RWED,RWED,RE)
ACLEDIT.TPU;1          [SYSTEM]              (RWED,RWED,RWED,RE)
ACLEDT$SECTION.TPU$SECTION;1
                       [SYSTEM]              (RWED,RWED,RWED,RE)
ACLEDTSHR.EXE;1        [SYSTEM]              (RWED,RWED,RWED,RE)
ADARTL.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
BASRTL.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
BASRTL2.EXE;1          [SYSTEM]              (RWED,RWED,RWED,RE)
CDDSHR.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
CLIMAC.REQ;1           [SYSTEM]              (RWED,RWED,RWED,RE)
COBRTL.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
CONVSHR.EXE;1          [SYSTEM]              (RWED,RWED,RWED,RE)
CRFSHR.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
DBGSSISHR.EXE;1        [SYSTEM]              (RWED,RWED,RWED,RE)
DCLTABLES.EXE;1        [SYSTEM]              (RWED,RWED,RWED,RE)
DCXSHR.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
DEBUG.EXE;1            [SYSTEM]              (RWED,RWED,RWED,RE)
DEBUGSHR.EXE;1         [SYSTEM]              (RWED,RWED,RWED,RE)
DELTA.EXE;1            [SYSTEM]              (RWED,RWED,RWED,RE)
DELTA.OBJ;1            [SYSTEM]              (RWED,RWED,RWED,RE)
DISMNTSHR.EXE;1        [SYSTEM]              (RWED,RWED,RWED,RE)
DTE_DF03.EXE;1         [SYSTEM]              (RWED,RWED,RWED,RE)
DTE_DF112.EXE;1        [SYSTEM]              (RWED,RWED,RWED,RE)
DTE_DMCL.EXE;1         [SYSTEM]              (RWED,RWED,RWED,RE)
DTKSHR.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
DYNSWITCH.EXE;1        [SYSTEM]              (RWED,RWED,RWED,RE)
EDTSHR.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
ENCRYPSHR.EXE;1        [SYSTEM]              (RWED,RWED,RWED,RE)
ERFCOMMON.EXE;1        [SYSTEM]              (RWED,RWED,RWED,RE)
ERFCTLSHR.EXE;1        [SYSTEM]              (RWED,RWED,RWED,RE)
ERFLIB.TLB;1           [SYSTEM]              (RWED,RWED,RWED,RE)
ERFSHR.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
ERFSHR2.EXE;1          [SYSTEM]              (RWED,RWED,RWED,RE)
EVE$SECTION.TPU$SECTION;1
                       [SYSTEM]              (RWED,RWED,RWED,RE)
FDLSHR.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
FORDEF.FOR;1           [SYSTEM]              (RWED,RWED,RWED,RE)
FORIOSDEF.FOR;1        [SYSTEM]              (RWED,RWED,RWED,RE)
FORRTL.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
FORRTL2.EXE;1          [SYSTEM]              (RWED,RWED,RWED,RE)
IMAGELIB.OLB;1         [SYSTEM]              (RWED,RWED,RWED,RE)
IMGDMP.EXE;1           [SYSTEM]              (RWED,RWED,RWED,RE)
```

```
LBRSHR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
LIB.MLB;2             [SYSTEM]          (RWED,RWED,RWED,RE)
LIB.REQ;1             [SYSTEM]          (RWED,RWED,RWED,RE)
LIBDEF.FOR;1          [SYSTEM]          (RWED,RWED,RWED,RE)
LIBRTL.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
LIBRTL2.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
MAILSHR.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
MAILSHR2.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
MOUNTSHR.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
MTHDEF.FOR;1          [SYSTEM]          (RWED,RWED,RWED,RE)
MTHRTL.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
NCS$LIBRARY.NLB;1     [SYSTEM]          (RWED,RWED,RWED,RE)
NCSSHR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
NMLSHR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
PASRTL.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
PLIRTL.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
PPLRTL.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
PPLSSISHR.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SCNRTL.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
SCRSHR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
SECURESHR.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SIGDEF.FOR;1          [SYSTEM]          (RWED,RWED,RWED,RE)
SMBSRVSHR.EXE;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SMGSHR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
SORTSHR.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
SPISHR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
STARLET.MLB;2         [SYSTEM]          (RWED,RWED,RWED,RE)
STARLET.OLB;2         [SYSTEM]          (RWED,RWED,RWED,RE)
STARLET.REQ;1         [SYSTEM]          (RWED,RWED,RWED,RE)
STARLETSD.TLB;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SUMSHR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
TFFSHR.REQ;1          [SYSTEM]          (RWED,RWED,RWED,RE)
TPAMAC.REQ;1          [SYSTEM]          (RWED,RWED,RWED,RE)
TPU$CCTSHR.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
TPU$DEBUG.TPU;1       [SYSTEM]          (RWED,RWED,RWED,RE)
TPUSHR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
TRACE.EXE;1           [SYSTEM]          (RWED,RWED,RWED,RE)
UISSHR.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
VAXCCURSE.OLB;1       [SYSTEM]          (RWED,RWED,RWED,RE)
VAXCRTL.EXE;1         [SYSTEM]          (RWED,RWED,RWED,RE)
VAXCRTL.OLB;1         [SYSTEM]          (RWED,RWED,RWED,RE)
VAXCRTLG.EXE;1        [SYSTEM]          (RWED,RWED,RWED,RE)
VAXCRTLG.OLB;1        [SYSTEM]          (RWED,RWED,RWED,RE)
VMSRTL.EXE;1          [SYSTEM]          (RWED,RWED,RWED,RE)
XFDEF.FOR;1           [SYSTEM]          (RWED,RWED,RWED,RE)

Total of 86 files.

Directory DB:[SYSO.SYSMGR]

ACCOUNTNG.DAT;1       [SYSTEM]          (RWED,RWED,RE,)
ALFMAINT.COM;1        [SYSTEM]          (RWED,RWED,RWED,RE)
CLUSTER_CONFIG.COM;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
EDTINI.EDT;1          [SYSTEM]          (RWED,RWED,RWED,RE)
EDTINI.TEMPLATE;1     [SYSTEM]          (RWED,RWED,RWED,RE)
ALFMAINT.COM;1        [SYSTEM]          (RWED,RWED,RWED,RE)
LIB$DT_STARTUP.COM;1
                      [SYSTEM]          (RWED,RWED,RWED,RE)
LOADNET.COM;1         [SYSTEM]          (RWED,RWED,RWED,RE)
LOGIN.COM;1           [SYSTEM]          (RWED,RWED,RWED,RE)
LOGIN.TEMPLATE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
LPA11STRT.COM;1       [SYSTEM]          (RWED,RWED,RWED,RE)
LTLOAD.COM;1          [SYSTEM]          (RWED,RWED,RWED,RE)
```

```
MAKEROOT.COM;1          [SYSTEM]            (RWED,RWED,RWED,RE)
NETCONFIG.COM;1         [SYSTEM]            (RWED,RWED,RWED,RE)
OPERATOR.LOG;3          [SYSTEM]            (RWED,RWED,RE,)
RTTLOAD.COM;1           [SYSTEM]            (RWED,RWED,RWED,RE)
SECAUDIT.COM;1          [SYSTEM]            (RWED,RWED,RWED,RE)
STARTNET.COM;1          [SYSTEM]            (RWED,RWED,RWED,RE)
SYCONFIG.COM;1          [SYSTEM]            (RWED,RWED,RWED,RE)
SYCONFIG.TEMPLATE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
SYPAGSWPFILES.COM;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
SYPAGSWPFILES.TEMPLATE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
SYSHUTDWN.COM;1         [SYSTEM]            (RWED,RWED,RWED,RE)
SYSHUTDWN.TEMPLATE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
SYLOGICALS.COM;1        [SYSTEM]            (RWED,RWED,RWED,RE)
SYLOGICALS.TEMPLATE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
SYLOGIN.COM;1           [SYSTEM]            (RWED,RWED,RWED,RE)
SYLOGIN.TEMPLATE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
SYSHUTDWN.COM;1         [SYSTEM]            (RWED,RWED,RWED,RE)
SYSHUTDWN.TEMPLATE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
SYSTARTUP_V5.COM;1      [SYSTEM]            (RWED,RWED,RWED,RE)
SYSTARTUP_V5.TEMPLATE;1
                        [SYSTEM]            (RWED,RWED,RWED,RE)
VMSIMAGES.DAT;1         [SYSTEM]            (RWED,RWED,,)
WELCOME.TEMPLATE;1      [SYSTEM]            (RWED,RWED,RWED,RE)
WELCOME.TXT;1           [SYSTEM]            (RWED,RWED,RWED,RE)

Total of 35 files.

Directory DB:[SYSO.SYSMSG]

ADAMSG.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
CLIUTLMSG.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
DBGTBKMSG.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
DBLRTLMSG.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
FILMNTMSG.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
NETWRKMSG.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
PASMSG.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
PLIMSG.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
PPLMSG.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
PRGDEVMSG.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
RPGMSG.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
SCNMSG.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
SHRIMGMSG.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
SYSMGTMSG.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
SYSMSG.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
TECOMSG.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)
TPUMSG.EXE;1            [SYSTEM]            (RWED,RWED,RWED,RE)
VAXCMSG.EXE;1           [SYSTEM]            (RWED,RWED,RWED,RE)

Total of 18 files.

Directory DB:[SYSO.SYSTEST]

TCNTRL.CLD;1            [SYSTEM]            (RWED,RWED,RWED,RE)
UETCLIGOO.COM;1         [SYSTEM]            (RWED,RWED,RWED,RE)
UETCLIGOO.DAT;1         [SYSTEM]            (RWED,RWED,RWED,RE)
UETCLIGOO.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
UETCOMSOO.EXE;1         [SYSTEM]            (RWED,RWED,RWED,RE)
```

```
UETDISK00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETDMPF00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETDNET00.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETDNET00.DAT;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETDR1W00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETDR7800.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETFORT01.DAT;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETFORT01.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETFORT02.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETFORT03.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETINIT00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETINIT01.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD00.DAT;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD02.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD03.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD04.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD05.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD06.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD07.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD08.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD09.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD10.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLOAD11.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETLPAK00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETMA7800.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETMEMY01.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETNETS00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETP.COM;1           [SYSTEM]          (RWED,RWED,RWED,RE)
UETPHAS00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETRSXFOR.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETSUPDEV.DAT;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETTAPE00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETTTYS00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)
UETUNAS00.EXE;1      [SYSTEM]          (RWED,RWED,RWED,RE)

Total of 39 files.

Directory DB:[SYS0.SYSUPD]

AUTOGEN.COM;1        [SYSTEM]          (RWED,RWED,RWED,RE)
BLISSREQ.TLR;1       [SYSTEM]          (RWED,RWED,RWED,RE)
BOOTBLDR.COM;1       [SYSTEM]          (RWED,RWED,RWED,RE)
BOOTUPD.COM;1        [SYSTEM]          (RWED,RWED,RWED,RE)
CONSCOPY.COM;1       [SYSTEM]          (RWED,RWED,RWED,RE)
CVTNAF.COM;1         [SYSTEM]          (RWED,RWED,RWED,RE)
CVTUAF.COM;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DECNET.TLR;1         [SYSTEM]          (RWED,RWED,RWED,RE)
DEVELOP.TLR;1        [SYSTEM]          (RWED,RWED,RWED,RE)
DXCOPY.COM;1         [SYSTEM]          (RWED,RWED,RWED,RE)
EXAMPLES.TLR;1       [SYSTEM]          (RWED,RWED,RWED,RE)
FILETOOLS.TLR;1      [SYSTEM]          (RWED,RWED,RWED,RE)
HELP.TLR;1           [SYSTEM]          (RWED,RWED,RWED,RE)
LIBDECOMP.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
LIBRARY.TLR;1        [SYSTEM]          (RWED,RWED,RWED,RE)
MANAGER.TLR;1        [SYSTEM]          (RWED,RWED,RWED,RE)
MISCTOOLS.TLR;1      [SYSTEM]          (RWED,RWED,RWED,RE)
QUEUES.TLR;1         [SYSTEM]          (RWED,RWED,RWED,RE)
REQUIRED.TLR;1       [SYSTEM]          (RWED,RWED,RWED,RE)
SETDEFBOO.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
SPKITBLD.COM;1       [SYSTEM]          (RWED,RWED,RWED,RE)
STABACKIT.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
SWAPFILES.COM;1      [SYSTEM]          (RWED,RWED,RWED,RE)
TEXTTOOLS.TLR;1      [SYSTEM]          (RWED,RWED,RWED,RE)
```

```
UETP.TLR;1              [SYSTEM]                (RWED,RWED,RWED,RE)
VMSINSTAL.COM;1         [SYSTEM]                (RWED,RWED,RWED,RE)
VMSKITBLD.COM;1         [SYSTEM]                (RWED,RWED,RWED,RE)
VMSKITBLD.DAT;1         [SYSTEM]                (RWED,RWED,RWED,RE)
VMSTAILOR.COM;1         [SYSTEM]                (RWED,RWED,RWED,RE)
VMSUPDATE.COM;1         [SYSTEM]                (RWED,RWED,RWED,RE)

Total of 30 files.

Directory DB:[SYSEXE]

SYSBOOT.EXE;1           [SYSTEM]                (RWED,RWED,RWED,RE)

Total of 1 file.

Directory DB:[SYSEXEMIN]

SYSBOOT.EXE;1           [SYSTEM]                (RWED,RWED,RWED,RE)

Total of 1 file.

Grand total of 14 directories, 539 files.
```

D Running VMS in a C2 Environment

Many of the security features provided by VMS are directed toward the requirements for a class C2 system, as defined in the *Department of Defense Trusted Computer System Evaluation Criteria*, published by the Department of Defense Computer Security Center (CSC-STD-001-83).

This appendix describes how some of the features of VMS relate to the C2 security model and notes some specific considerations for operating a VMS system within the C2 framework. Since the terminology used in this appendix is drawn from the evaluation criteria, you should be familiar with the government document before using this appendix.

The Trusted Computing Base

The Trusted Computing Base (TCB) provided by VMS encompasses much of the operating system. It includes the entire executive and file system, all other system components that execute in inner access modes (such as device drivers, RMS, and DCL), most system programs installed with privilege, and a variety of other utilities used by system managers to maintain data relevant to the TCB.

The objects for which VMS provides full C2-style protection are files and directories that are accessible through normal file access techniques or through global sections. VMS also provides access controls of varying levels for other objects such as devices, logical name tables, and queues.

Protecting the TCB

The code and data that make up the VMS TCB reside in files and, in part, in the address space of the running operating system. Their integrity is protected by the use of file access controls and memory page protection. Memory page protection is set up by VMS as it executes and is normally not of concern to the system manager. The files containing the TCB are by default correctly protected when VMS is installed; however, the protection can be altered by sufficiently privileged users. Appendix C of this handbook describes the correct file protection of all VMS components.

Certain privileges allow the holder to bypass normal file and memory access controls directly or indirectly and, therefore, must not be granted to persons other than the system manager, security officer, or other trusted persons. These privileges are described in detail in Appendix A of this handbook. Table 5-5 in this manual summarizes the privileges and categorizes them in terms of their sensitivity.

Privileges in the FILES and ALL categories allow the holder to violate the integrity of the TCB. Privileges in the SYSTEM category allow the holder to interfere with normal system operation and cause denial of service; however, they do not allow the holder to actually violate object access controls.

Some privileges in the SYSTEM category also permit certain forms of deception that could ultimately result in violations of access controls. Privileges in the DEVOUR and GROUP categories permit the holder to consume resources without limit, thereby causing possible denial of service and interference with the operations of others in the same group. The

GRPPRV privilege, in particular, permits the holder to violate normal access controls within that holder's group.

Individual Accountability

The proper use of user names, UICs, and passwords ensures that individual accountability is enforced by VMS. The following practices and features, however, result in the loss of individual accountability and must not be used in a C2 environment:

1. Do not assign the same UIC to more than one user. The UIC is used as the universal internal user identifier; therefore, unique UICs must be assigned to all users.
2. Do not allow open accounts. Lack of a password makes an account available to all users aware of its identity. The system manager can prevent open accounts by never setting null passwords with AUTHORIZE and by ensuring that all accounts are set up with a nonzero minimum password length.
3. Do not allow group accounts. Individual accountability is lost when more than one person shares an account. Each user must be given a unique account.
4. Do not enable autologin. Autologin associates an account with a particular terminal instead of a particular person and, therefore, causes a loss of individual accountability.
5. Do not initiate network proxy accounts for groups. In order to preserve individual accountability, each individual in a network must be given a unique network proxy account on each node to which that user has access. Assign the same user name and UIC on all applicable nodes. Then set up individual proxies among the corresponding accounts.

Object Protection and Reuse

File and directory protection is described extensively in Chapter 4. A series of mechanisms, described in Section 4.5, provides useful default protection for newly created objects.

Reuse of system memory pages is protected by the memory management subsystem and cannot be defeated. Reuse of disk blocks is protected by the highwater marking and erase-on-delete features, which are described in Section 5.6. To conform to C2 criteria, all system disk volumes must have highwater marking enabled, which is the default.

VMS considers magnetic tapes to be single user devices. Tape protection is available only at volume level. An entire volume can be assigned ownership and protection, but the individual files it contains cannot. Because of this policy, VMS provides no protection against reuse of tape. Tapes that are recycled to new users must be erased externally by operations personnel.

Protection of the Audit Trail

The security audit trail is recorded in the operator log file and on terminals enabled as security operators. The operator log is normally protected against reading or modification by unauthorized users.

To make sure evidence of system penetration cannot be fully erased by a malicious user, further protect of the audit log by adopting the following measures:

1 In a physically secure location, place a hardcopy terminal enabled as a security operator.

2 Protect the audit log file with the following audit measures:

   a. Enable, at minimum, audits on ACL and audit events with the following command:

```
$ SET AUDIT /ALARM /ENABLE=(ACL,AUDIT)
```

   b. Place on the operator log file an ACL entry (ACE) that enables auditing of all accesses for modification and deletion:

```
$ SET ACL SYS$MANAGER:OPERATOR.LOG -
_$ /ACL=(ALARM=SECURITY,ACCESS=WRITE+DELETE+CONTROL+SUCCESS)
```

   These audits ensure that any attempt to tamper with the audit log will result either in the system audit controls left obviously turned off, or with one last "footprint" in the audit log. While these measures are not completely secure, circumventing them requires extensive programming.

Auditing Actions of a System Operator or Administrator

Actions taken by trusted users of the system (operators, administrators, and security officers) can be audited by the enforced use of terminal session auditing as described in Section 5.8.5. Attempts to defeat the auditing can be detected by taking measures similar to those used to protect the audit log:

1 Enable auditing of authorization modifications.

2 Place ACL entries on the captive login command procedures and the directories containing them to detect modification of the procedures.

Documentation

The *Guide to VMS System Security* and any applicable reference documentation make up the *Trusted Facility Manual* for use by the system administrator. The first four chapters of this guide constitute the *Security Features User's Guide* and should be made available to all system users.

Physical Security

Physical and environmental security are critical to the secure operation of the system. Preventing theft of media and output is an obvious consideration. In addition, the console terminal must always be physically secured because it controls operation of the CPU and, consequently, operation of the system.

Configuration Guidelines

The security features described in this guide apply to most VAX configurations. They are supported by all VAX CPUs, including MicroVAX CPUs, and apply to all supported mass storage and communications devices.

VAXcluster configurations also fully support the security features. A VAXcluster is considered a single security and management domain and normally operates with a shared authorization database. For more information, please refer to the *VMS VAXcluster Manual.*

Running VMS in a C2 Environment

VMS does not meet the needs of the C2 requirements in configurations in which an MA780 shared memory subsystem is used as a shared memory among two or more independent processors. The communications objects in the shared memory (common event flag clusters, mailboxes, and global sections) do not support access control lists or security auditing. However, when a VAX-11/782 dual processor system uses the MA780 as the main memory, all security features are fully supported.

The evaluation criteria do not address network operation. However, when connected to a DECnet network, VMS provides security commensurate with the security of the base operating system if the following restrictions are met:

- All operating systems connected to the network are VMS systems or systems of equivalent security and are systems administered in a secure manner.
- Default accounts are not provided for file and general task access. Limiting access to explicit access control strings and proxy access preserves individual accountability.
- Communications lines are secured from wiretaps by use of link encryption devices or by physical security.

E Alarm Messages

This appendix describes alarm messages that result from auditing various system events.

E.1 Alarms Auditing Access to Files and Global Sections

You can audit successful or unsuccessful access to a file or global section by specifying the FILE_ACCESS keyword with the /ENABLE qualifier of the SET AUDIT command. READ, WRITE, EXECUTE, DELETE, or CONTROL access modes can be audited. You can also audit successful access to a file or global section through the use of GRPPRV, READALL, SYSPRV, or BYPASS privilege.

In addition to the information contained in all alarm messages, alarms auditing access to files and global sections contain the following:

- Name of the file or global section accessed
- Mode of access
- Image used to access the file or global section
- Privileges used to access the file or global section

The following alarm messages are examples of the file and global section access alarms.

```
%%%%%%%%%%  OPCOM  15-JUN-1988 12:19:49.95  %%%%%%%%%%
Security alarm on LASSIE / Successful file access
        Time:           15-JUN-1988 12:19:49.94
        PID:            26C0062B
        User Name:      WILSON
        Image:          LASSIE$DMA0:[SYSTEM.SYSEXE]:TYPE.EXE
        File:           _LASSIE$DMA0:[TIMMY]PRIVILEGE.CMD;1
        Mode:           READ
        Privs Used:     (None)

%%%%%%%%%%  OPCOM  15-JUN-1988 12:20:14.61  %%%%%%%%%%
Security alarm on LASSIE / File access failure
        Time:           15-JUN-1988 12:20:14.60
        PID:            26C0062B
        User Name:      WILSON
        Image:          LASSIE$DMA0:[SYSTEM.SYSEXE]:TYPE.EXE
        File:           _LASSIE$DMA0:[TIMMY]PRIVILEGE.CMD;1
        Mode:           READ

%%%%%%%%%%  OPCOM  15-JUN-1988 12:22:01.17  %%%%%%%%%%
Security alarm on LASSIE / Successful global section access
        Time:           15-JUN-1988 12:22:01.16
        PID:            00000112
        Name:           SYSTEM
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]MAIL.EXE
        Global Section: SMG$TERMTABLE
        File:           _LASSIE$DMA0:[SYS0.][SYSEXE]TERMTABLE.EXE;1
        Mode:           READ
        Privs Used:     (None)
```

E.1 Alarms Auditing Access to Files and Global Sections

```
%%%%%%%%%%  OPCOM  18-APR-1988 10:11:40.21  %%%%%%%%%%
Security alarm on VENUS / Global section access failure
        Time:           18-APR-1988 10:11:40.20
        PID:            000000A5
        User Name:      SYSTEM
        Image:          VENUS$DMA0:[SYS0.][SYSMGR]PRIV_USERS.EXE;3
        Global Section: ACLEDT_002
        File:           _VENUS$DMA0:[SYS0.][SYSEXE]ACLEDT.EXE;1
        Mode:           READ

%%%%%%%%%%  OPCOM  15-JUN-1988 12:19:49.95  %%%%%%%%%%
Security alarm on LASSIE / Successful file access
        Time:           15-JUN-1988 12:19:49.94
        PID:            26C0062B
        User Name:      WILSON
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]TYPE.EXE
        File:           _LASSIE$DMA0:[TIMMY.CMDS]PRIVILEGE.CMD;1
        Mode:           READ
        Privs Used:     BYPASS
```

E.2 Alarms Requested by an ACL

You can audit successful or unsuccessful access to a file. To do so, add an ALARM_JOURNAL ACE to a file's ACL, and then enable ACL alarms by specifying the ACL keyword with the /ENABLE qualifier of the SET AUDIT command. For example, the following ACE in a file's ACL requests that an alarm occur whenever the file is successfully read:

```
(ALARM_JOURNAL=SECURITY,ACCESS=SUCCESS+READ)
```

The ACL alarm has no effect unless it is enabled with the following command:

```
$ SET AUDIT/ALARM/ENABLE=(ACL)
```

READ, WRITE, EXECUTE, DELETE, or CONTROL modes can be audited. In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Name of the file or global section accessed
- Mode of access
- Image used to access the file or global section
- Privileges used to access the file or global section

The following alarm messages are examples of alarms requested by an ACL:

```
%%%%%%%%%%  OPCOM  15-JUN-1988 12:19:49.95  %%%%%%%%%%
Security alarm on LASSIE / Successful file access
        Time:           15-JUN-1988 12:19:49.94
        PID:            26C0062B
        User Name:      MENACE
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]TYPE.EXE
        File:           _LASSIE$DMA0:[TIMMY]PRIVATE.LIS;1
        Mode:           READ
        Privs Used:     (None)

%%%%%%%%%%  OPCOM  15-JUN-1988 12:20:14.61  %%%%%%%%%%
Security alarm on LASSIE / File access failure
        Time:           15-JUN-1988 12:20:14.60
        PID:            26C0062B
        User Name:      MENACE
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]TYPE.EXE
        File:           _LASSIE$DMA0:[TIMMY]PRIVATE.LIS;1
        Mode:           READ
```

E.3 Alarms Auditing INSTALL Operations

You can audit the use of the Install Utility (to install an image or to remove an installed image) by specifying the INSTALL keyword with the /ENABLE qualifier of the SET AUDIT command. In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Type of INSTALL operation
- Name of the image affected by the INSTALL operation
- Flags set by INSTALL operation
- Privileges used in the INSTALL operation

The following alarm messages are examples of alarms resulting from INSTALL operations.

```
%%%%%%%%%%  OPCOM  15-JUN-1988 12:37:49.69  %%%%%%%%%%
Security alarm on LASSIE / Installed file addition
        Time:           15-JUN-1988 12:37:49.68
        PID:            00000113
        User Name:      SYSTEM
        File:           LASSIE$DMA0:[SYS0.][SYSEXE]NCP.EXE;9
         INSTALL Flags: OPEN HEADER_RESIDENT

%%%%%%%%%%  OPCOM  15-JUN-1988 12:37:54.20  %%%%%%%%%%
Security alarm on LASSIE / Installed file removal
        Time:           15-JUN-1988 12:37:54.18
        PID:            00000113
        User Name:      SYSTEM
        File:           LASSIE$DMA0:[SYS0.][SYSEXE]NCP.EXE;9
        INSTALL Flags:  OPEN HEADER_RESIDENT

%%%%%%%%%%  OPCOM   3-MAY-1988 10:12:19.18  %%%%%%%%%%
Security alarm on VENUS / Installed file addition
        Time:           03-MAY-1988 10:12:19.11
        PID:            24E00A03
        User Name:      SMITH
        File:           DUA8:[SYS8.SYSCOMMON.SYSEXE]SDA.EXE;1
        INSTALL Flags:  PRIVILEGED
        Privileges:     CMKRNL
```

```
%%%%%%%%%%  OPCOM   3-MAY-1988 10:12:27.84  %%%%%%%%%%
Security alarm on VENUS / Installed file removal
        Time:           03-MAY-1988 10:12:27.83
        PID:            24E00A03
        User Name:      SMITH
        File:           DUA8:[SYS8.SYSCOMMON.SYSEXE]SDA.EXE;1
        INSTALL Flags:  PRIVILEGED
        Privileges:     CMKRNL
```

E.4 Alarms Resulting from Modifications to the Rights Database

You can audit any changes made to the rights database by specifying the AUTHORIZATION keyword with the /ENABLE qualifier of the SET AUDIT command. Following are the types of changes that you can audit:

- Creation of a new rights database
- Addition of an identifier
- Removal of an identifier
- Modification of an identifier
- Granting of an identifier to a holder
- Revoking an identifier from a holder
- Modification of a holder

In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Image used to modify the rights database
- Change made to the rights database

The following alarm messages are examples of alarms resulting from modification of the rights database.

```
%%%%%%%%%%  OPCOM  25-APR-1988 10:42:34.42  %%%%%%%%%%
Security alarm on VENUS / Rights database created
        Time:           25-APR-1988 10:42:34.39
        PID:            22200150
        User Name:      SMITH
        Image:          DUA8:[SYS8.][SYSCOMMON.SYSEXE]AUTHORIZE.EXE
        File:           SYS$SYSTEM:RIGHTSLIST.DAT;1

%%%%%%%%%%  OPCOM  15-JUN-1988 12:26:44.60  %%%%%%%%%%
Security alarm on LASSIE / Identifier added
        Time:           15-JUN-1988 12:26:44.01
        PID:            00000113
        User Name:      SYSTEM
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]AUTHORIZE.EXE
        ID name:        TIMMY
        ID value:       %X80010000
        Id attributes:  (None)

%%%%%%%%%%  OPCOM  15-JUN-1988 12:33:23.06  %%%%%%%%%%
Security alarm on LASSIE / Identifier removed
        Time:           15-JUN-1988 12:33:23.05
        PID:            00000113
        User Name:      SYSTEM
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]AUTHORIZE.EXE
```

```
        ID name:        MASTERS
        ID value:       %X80010000

%%%%%%%%%%  OPCOM  15-JUN-1988 12:27:17.44  %%%%%%%%%%
Security alarm on LASSIE / Identifier modified
        Time:           15-JUN-1988 12:27:17.36
        PID:            00000113
        User Name:      SYSTEM
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]AUTHORIZE.EXE
        ID name:        ROBINSON
        ID value:       %X80010000
        Id attributes:  RESOURCE

%%%%%%%%%%  OPCOM   6-MAY-1988 10:14:59.42  %%%%%%%%%%
Security alarm on VENUS / Identifier modified
        Time:           06-MAY-1988 10:14:59.32
        PID:            20A00385
        User Name:      SMITH
        Image:          VENUS$DUB44:[SYS0.][SYSCOMMON.SYSEXE]AUTHORIZE.EXE
        ID name:        DEMILO
        ID value:       %X800186A0
        New ID value:   %X800186A1

%%%%%%%%%%  OPCOM   6-MAY-1988 10:15:12.02  %%%%%%%%%%
Security alarm on VENUS / Identifier modified
        Time:           06-MAY-1988 10:15:12.02
        PID:            20A00385
        User Name:      SMITH
        Image:          VENUS$DUB44:[SYS0.][SYSCOMMON.SYSEXE]AUTHORIZE.EXE
        ID name:        BONNELL
        ID value:       %X800186A1
        New ID name:    SMITH

%%%%%%%%%%  OPCOM  15-JUN-1988 12:28:19.54  %%%%%%%%%%
Security alarm on LASSIE / Identifier granted
        Time:           15-JUN-1988 12:28:19.52
        PID:            00000113
        User Name:      SYSTEM
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]AUTHORIZE.EXE
        ID name:        PERSONNEL
        ID value:       %X80010000
        Holder name:    PERKINS
        Holder UIC:     [214,202]
        Id attributes:  (None)

%%%%%%%%%%  OPCOM  15-JUN-1988 12:28:49.61  %%%%%%%%%%
Security alarm on LASSIE / Identifier revoked
        Time:           15-JUN-1988 12:28:49.60
        PID:            00000113
        User Name:      SYSTEM
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]AUTHORIZE.EXE
        ID name:        PERSONNEL
        ID value:       %X80010000
        Holder name:    MANSON
        Holder UIC:     [214,210]

%%%%%%%%%%  OPCOM  15-JUN-1988 12:28:19.54  %%%%%%%%%%
Security alarm on LASSIE / ID holder modified
        Time:           15-JUN-1988 12:28:19.52
        PID:            00000113
        User Name:      SYSTEM
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]AUTHORIZE.EXE
        ID name:        PAYROLL
        ID value:       %X80010000
```

```
Holder name:     ARNOLD
Holder UIC:      [220,133]
Id attributes:   RESOURCE
```

E.5 Alarms Resulting from Changes to SYSUAF or NETPROXY

Specifying the AUTHORIZATION keyword with the /ENABLE qualifier of the SET AUDIT command enables auditing of changes made to the system UAF or network proxy authorization file in addition to auditing changes to the rights database. Following are the types of changes that you can audit:

- Adding a system UAF record
- Deleting a system UAF record
- Changing a system UAF record
- Copying a system UAF record
- Renaming a system UAF record
- Adding network proxy access
- Deleting network proxy access
- Modifying network proxy access

In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Record that was modified
- Change that was made

The following alarm messages are examples of alarms resulting from modification of the system or network UAF.

```
%%%%%%%%%%  OPCOM  15-JUN-1988 12:27:37.03  %%%%%%%%%%
Security alarm on LASSIE / System UAF record addition
        Time:           15-JUN-1988 12:27:36.97
        PID:            00000113
        User Name:      SYSTEM
        Rec Add:        COOPER
        Fields Mod:     FLAGS         PWDLIFETIME

%%%%%%%%%%  OPCOM  15-JUN-1988 12:33:28.36  %%%%%%%%%%
Security alarm on LASSIE / System UAF record deletion
        Time:           15-JUN-1988 12:33:28.35
        PID:            00000113
        User Name:      SYSTEM
        Rec Del:        MELVILLE

%%%%%%%%%%  OPCOM  15-JUN-1988 12:27:52.26  %%%%%%%%%%
Security alarm on LASSIE / System UAF record modification
        Time:           15-JUN-1988 12:27:52.25
        PID:            23C00155
        User Name:      MENACE
        Rec Mod:        GOWER
        Fields Mod:     PRIVILEGES

%%%%%%%%%%  OPCOM  15-JUN-1988 12:32:28.50  %%%%%%%%%%
Security alarm on LASSIE / System UAF record copied
```

```
        Time:           15-JUN-1988 12:32:28.49
        PID:            00000113
        User Name:      SYSTEM
        Rec Add:        MUTT
        From Name:      JEFF
        Fields Mod:     (None)

%%%%%%%%%%  OPCOM  15-JUN-1988 12:32:40.48  %%%%%%%%%%
Security alarm on LASSIE / System UAF record renamed
        Time:           15-JUN-1988 12:32:40.47
        PID:            00000113
        User Name:      SYSTEM
        New Name:       ALIAS
        Old Name:       SILAS
        Fields Mod:     (None)

%%%%%%%%%%  OPCOM  15-JUN-1988 12:34:13.84  %%%%%%%%%%
Security alarm on LASSIE / Network UAF record addition
        Time:           15-JUN-1988 12:34:13.77
        PID:            00000113
        User Name:      SYSTEM
        Rec Add:        VENUS::SYSTEM       SYSTEM

%%%%%%%%%%  OPCOM  15-JUN-1988 12:36:23.96  %%%%%%%%%%
Security alarm on LASSIE / Network UAF record deletion
        Time:           15-JUN-1988 12:36:23.95
        PID:            00000113
        User Name:      SYSTEM
        Rec Del:        GEORGE::SYSTEM       SYSTEM

%%%%%%%%%%  OPCOM  18-APR-1988 09:50:05.12  %%%%%%%%%%
Security alarm on VENUS / Network UAF record modification
        Time:           18-APR-1988 09:50:04.11
        PID:            000000A5
        User Name:      SYSTEM
        New Rec:        MARS::USER          SUESS
        Old Rec:        MARS::USER          WIMBLY
```

E.6 Alarms Resulting from Password Changes

The AUTHORIZATION keyword specified with the /ENABLE qualifier of the SET AUDIT command also enables auditing of changes to a user or the system password. In addition to the information contained in all alarm messages, this type of alarm specifies which password was changed.

The following alarm messages are examples of alarms resulting from modification of the system or network UAF.

```
%%%%%%%%%%  OPCOM  15-JUN-1988 12:27:52.26  %%%%%%%%%%
Security alarm on LASSIE / System UAF record modification
        Time:           15-JUN-1988 12:27:52.25
        PID:            20C00133
        User Name:      MENACE
        Rec Mod:        DENNIS
        Fields Mod:     PASSWORD

%%%%%%%%%%  OPCOM   3-MAY-1988 14:17:33.98  %%%%%%%%%%
Security alarm on VENUS / System UAF record modification
        Time:           03-MAY-1988 14:17:33.90
        PID:            24E00A03
        User Name:      MENACE
        Rec Mod:        <SYSTEM-PASSWORD>
        Fields Mod:     PASSWORD
```

E.7 Break-In Attempt Alarms

You can audit break-in attempts by specifying the BREAKIN keyword with the /ENABLE qualifier of the SET AUDIT command. You can audit the DIALUP, LOCAL, REMOTE, NETWORK and DETACHED break-in types.

In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Type of break-in attempt
- Password used
- Device used
- Origin of attempt if the break-in type was REMOTE or NETWORK
- Parent user name if the break-in type was DETACHED

The following alarm messages are examples of the break-in attempt alarms.

```
%%%%%%%%%%  OPCOM  15-JUN-1988 14:33:20.69  %%%%%%%%%%
Security alarm on LASSIE / Dialup interactive breakin detection
        Time:           15-JUN-1988 14:33:20.67
        PID:            00000052
        User Name:      SNIDELY
        Password:       EASYDOESIT
        Dev Name:       _TTA1:

%%%%%%%%%%  OPCOM  15-APR-1988 14:32:58.79  %%%%%%%%%%
Security alarm on LASSIE / Local interactive breakin detection
        Time:           15-APR-1988 14:32:58.77
        PID:            00000051
        User Name:      SNIDELY
        Password:       MUMBLE
        Dev Name:       _TTA1:

%%%%%%%%%%  OPCOM  17-APR-1988 14:54:34.87  %%%%%%%%%%
Security alarm on LASSIE / Remote interactive breakin detection
        Time:           17-APR-1988 14:54:34.83
        PID:            0000005D
        User Name:      FAGAN
        Password:       LIGHTLY
        Dev Name:       _RTA1:
        Source:         2.218  VENUS::SYSTEM

%%%%%%%%%%  OPCOM  17-APR-1988 14:55:28.51  %%%%%%%%%%
Security alarm on LASSIE / Network breakin detection
        Time:           17-APR-1988 14:55:28.50
        PID:            0000005E
        User Name:      DECNET
        Password:       FARFARAWAY
        Source:         2.218  VENUS::SYSTEM

%%%%%%%%%%  OPCOM  18-APR-1988 16:14:59.72  %%%%%%%%%%
Security alarm on LASSIE / Detached process breakin detection
        Time:           18-APR-1988 16:14:59.60
        PID:            00000162
        User Name:      ARTFUL
        Password:       DODGER
        Parent U.N.:    SYSTEM
```

E.8 Login Alarms

You can audit successful logins by specifying the LOGIN keyword with the /ENABLE qualifier of the SET AUDIT command. You can audit BATCH, DIALUP, LOCAL, REMOTE, NETWORK, SUBPROCESS and DETACHED login types.

In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Type of login
- Device used
- Origin of the login if it was REMOTE or NETWORK
- Parent PID if the login was SUBPROCESS
- Parent user name if the login was DETACHED

The following alarm messages are examples of successful login alarms.

```
%%%%%%%%%%  OPCOM  17-APR-1988 14:57:19.25  %%%%%%%%%%
Security alarm on TIGER / Batch process login
        Time:           17-APR-1988 14:57:19.24
        PID:            320B005F
        User Name:      MENACE

%%%%%%%%%%  OPCOM  15-APR-1988 14:34:33.68  %%%%%%%%%%
Security alarm on TIGER / Dialup interactive login
        Time:           15-APR-1988 14:34:33.67
        PID:            00030055
        User Name:      GRAHAM
        Dev Name:       _TTA1:

%%%%%%%%%%  OPCOM  15-JUN-1988 12:53:01.56  %%%%%%%%%%
Security alarm on LASSIE / Local interactive login
        Time:           15-JUN-1988 12:53:01.55
        PID:            03500124
        User Name:      JUNE
        Dev Name:       _LTA7:

%%%%%%%%%%  OPCOM  15-JUN-1988 12:52:38.85  %%%%%%%%%%
Security alarm on LASSIE / Remote interactive login
        Time:           15-JUN-1988 12:52:38.84
        PID:            00000123
        User Name:      SYSTEM
        Dev Name:       _RTA1:
        Source:         2.91  LASSIE::SYSTEM

%%%%%%%%%%  OPCOM  15-JUN-1988 12:50:25.93  %%%%%%%%%%
Security alarm on LASSIE / Network login
        Time:           15-JUN-1988 12:50:25.92
        PID:            00000121
        User Name:      SYSTEM
        Source:         2.58  TIGER::2E0004A4

%%%%%%%%%%  OPCOM  15-JUN-1988 12:49:54.92  %%%%%%%%%%
Security alarm on LASSIE / Subprocess login
        Time:           15-JUN-1988 12:49:54.91
        PID:            00000120
        User Name:      ADAM
        Parent PID:     00000113
```

```
%%%%%%%%%%  OPCOM  17-APR-1988 17:08:08.34  %%%%%%%%%%
Security alarm on TIGER / Detached process login
        Time:           17-APR-1988 17:08:08.31
        PID:            00000093
        User Name:      ISSAC
        Parent U.N.:    ABRAHAM
```

E.9 Login Failure Alarms

You can audit login failures by specifying the LOGFAILURE keyword with the /ENABLE qualifier of the SET AUDIT command. You can audit the BATCH, DIALUP, LOCAL, REMOTE, NETWORK, SUBPROCESS and DETACHED login failure types.

In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Type of login
- Device used
- Status detailing the reason for the failure
- Origin of the login if it was REMOTE or NETWORK
- Parent PID if the login was SUBPROCESS
- Parent user name if the login was DETACHED

The following alarm messages are examples of login failure alarms.

```
%%%%%%%%%%  OPCOM  17-APR-1988 15:03:30.86  %%%%%%%%%%
Security alarm on TIGER / Batch process login failure
        Time:       17-APR-1988 15:03:30.78
        PID:        00030060
        User Name:  RUBENS
        Status:     %LOGIN-F-BADDAY, you are not authorized to login today

%%%%%%%%%%  OPCOM  15-APR-1988 14:35:30.77  %%%%%%%%%%
Security alarm on TIGER / Dialup interactive login failure
        Time:       15-APR-1988 14:35:30.47
        PID:        00000057
        User Name:  LILY
        Status:     %LOGIN-F-NOSUCHUSER, no such user
        Dev Name:   _TTA1:

%%%%%%%%%%  OPCOM  15-JUN-1988 12:46:36.82  %%%%%%%%%%
Security alarm on LASSIE / Local interactive login failure
        Time:       15-JUN-1988 12:46:36.80
        PID:        0000011C
        User Name:  TIMMY
        Status:     %LOGIN-F-NOSUCHUSER, no such user
        Dev Name:   _LTA6:

%%%%%%%%%%  OPCOM  15-JUN-1988 12:40:50.40  %%%%%%%%%%
Security alarm on LASSIE / Remote interactive login failure
        Time:       15-JUN-1988 12:40:50.24
        PID:        00000115
        User Name:  THOMPSON
        Status:     %LOGIN-F-NOSUCHUSER, no such user
        Dev Name:   _RTA1:
        Source:     2.91  LASSIE::SYSTEM
```

```
%%%%%%%%%%  OPCOM  15-JUN-1988 12:48:43.50  %%%%%%%%%%
Security alarm on LASSIE / Network login failure
        Time:        15-JUN-1988 12:48:43.49
        PID:         0000011D
        User Name:   DECNET
        Status:      %LOGIN-F-NOSUCHUSER, no such user
        Source:      2.58  MILOS::SMITH

%%%%%%%%%%  OPCOM  18-APR-1988 17:07:54.60  %%%%%%%%%%
Security alarm on LASSIE / Subprocess login failure
        Time:        18-APR-1988 17:07:54.59
        PID:         00000195
        User Name:   RANGER
        Status:      %LOGIN-F-OUTPUTERR, error opening primary
                     output file SYS$OUTPUT
        Parent PID:  0000018D

%%%%%%%%%%  OPCOM  15-JUN-1988 12:49:30.69  %%%%%%%%%%
Security alarm on LASSIE / Detached process login failure
        Time:        15-JUN-1988 12:49:30.68
        PID:         0000011E
        User Name:   PUPPY
        Status:      %SYSTEM-F-NOLOGNAM, no logical name match
        Parent U.N.: DOGGIE
```

E.10 Logout Alarms

You can audit logouts by specifying the LOGOUT keyword with the /ENABLE qualifier of the SET AUDIT command. You can audit BATCH, DIALUP, LOCAL, REMOTE, NETWORK, SUBPROCESS and DETACHED logout types.

In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Type of logout
- Device used
- Origin of the login if it was REMOTE or NETWORK
- Parent PID if the login was SUBPROCESS

The following alarm messages are examples of logout alarms:

```
%%%%%%%%%%  OPCOM  17-APR-1988 14:57:21.11  %%%%%%%%%%
Security alarm on TIGER / Batch process logout
        Time:           17-APR-1988 14:57:21.10
        PID:            0000005F
        User Name:      LILY

%%%%%%%%%%  OPCOM  15-APR-1988 14:35:15.78  %%%%%%%%%%
Security alarm on TIGER / Dialup interactive logout
        Time:           15-APR-1988 14:35:15.77
        PID:            00000056
        User Name:      LILY
        Dev Name:       _TTA1:

%%%%%%%%%%  OPCOM  15-JUN-1988 12:53:07.09  %%%%%%%%%%
Security alarm on LASSIE / Local interactive logout
        Time:           15-JUN-1988 12:53:07.08
        PID:            00000124
        User Name:      TIMMY
        Dev Name:       _LTA7:

%%%%%%%%%%  OPCOM  15-JUN-1988 12:52:41.23  %%%%%%%%%%
Security alarm on LASSIE / Remote interactive logout
        Time:           15-JUN-1988 12:52:41.22
        PID:            00000123
        User Name:      TIMMY
        Dev Name:       _RTA1:
        Source:         2.91  LASSIE::TIMMY

%%%%%%%%%%  OPCOM  17-APR-1988 14:44:22.23  %%%%%%%%%%
Security alarm on VENUS / Network logout
        Time:           17-APR-1988 14:44:22.22
        PID:            00000058
        User Name:      DEMILO
        Source:         2.91  LASSIE::SYSTEM

%%%%%%%%%%  OPCOM  15-JUN-1988 12:49:59.49  %%%%%%%%%%
Security alarm on LASSIE / Subprocess logout
        Time:           15-JUN-1988 12:49:59.48
        PID:            00000120
        User Name:      TIMMY
        Parent PID:     00000113

%%%%%%%%%%  OPCOM  17-APR-1988 17:08:09.09  %%%%%%%%%%
Security alarm on VENUS / Detached process logout
        Time:           17-APR-1988 17:08:09.08
        PID:            00000093
        User Name:      DEMILO
```

E.11 Volume Mount and Dismount Alarms

You can audit login failures by specifying the MOUNT keyword with the /ENABLE qualifier of the SET AUDIT command. In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Image used to mount the volume
- Device used
- Log file recording the operation
- Volume name, UIC, and protection

- Flags set during the operation

The following alarm message is an example of a mount alarm.

```
%%%%%%%%%%  OPCOM  15-JUN-1988 12:57:00.21  %%%%%%%%%%
Security alarm on LASSIE / Volume mount
        Time:           15-JUN-1988 12:56:59.93
        PID:            00000113
        User Name:      SYSTEM
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]VMOUNT.EXE;1
        Dev Name:       _LASSIE$CSA1:
        Log Name:       DISK$CONSOLE
        Vol Name:       CONSOLE
        Vol UIC:        [1,4]
        Vol Pro:        S:RWED,O:RWED,G,W
        Mount Flags:    FOREIGN    MESSAGE    SYSTEM
```

The MOUNT keyword specified with the /ENABLE qualifier of the SET AUDIT command also enables auditing of dismount operations. In addition to the information contained in all alarm messages, this type of alarm contains the following:

- Image used to dismount the volume
- Device used
- Log file recording the operation
- Volume name
- Flags set during the operation

The following alarm message is an example of a dismount alarm.

```
%%%%%%%%%%  OPCOM  15-JUN-1988 12:57:06.78  %%%%%%%%%%
Security alarm on LASSIE / Volume dismount
        Time:           15-JUN-1988 12:57:06.77
        PID:            00000113
        User Name:      SYSTEM
        Image:          LASSIE$DMA0:[SYS0.][SYSEXE]DISMOUNT.EXE;1
        Dev Name:       _LASSIE$CSA1:
        Log Name:       DISK$CONSOLE
        Vol Name:       CONSOLE
        Mount Flags:    (None)
```

E.12 Alarms Resulting from Execution of SET AUDIT Command

You can enable auditing of any execution of the SET AUDIT command by specifying the AUDIT keyword with the /ENABLE qualifier of the SET AUDIT command. By using this type of alarm, you audit the use of alarms.

The following alarm messages are examples of audit alarms:

E.12 Alarms Resulting from Execution of SET AUDIT Command

```
%%%%%%%%%%  OPCOM   3-MAY-1988 10:11:19.42  %%%%%%%%%%
Security alarm on VENUS / Security audit alarms enabled
        Time:           03-MAY-1988 10:11:19.41
        PID:            24E00A03
        User Name:      SMITH
        Audits Modified:
            INSTALL

%%%%%%%%%%  OPCOM   3-MAY-1988 10:12:58.84  %%%%%%%%%%
Security alarm on VENUS / Security audit alarms disabled
        Time:           03-MAY-1988 10:12:58.83
        PID:            24E00A03
        User Name:      SMITH
        Audits Modified:
            INSTALL
            AUDIT
```

Glossary

Access Control List: A list that defines the kinds of access to be granted or denied to users of an object. Access control lists can be created for objects such as files, devices, and mailboxes. Each access control list consists of one or more entries known as access control list entries.

Access Control List Entry: An entry in an access control list. Access control list entries may specify identifiers and the access rights to be granted or denied the holders of the identifiers, default protection for directories, or security alarm details. Access control lists for each object can hold many entries, limited only by overall space and performance considerations.

ACE: See *Access Control List Entry*

ACL Editor: A VMS utility that helps users create and maintain access control lists.

ACL: See *Access Control List*

Alarm: See *Security Alarm*

Alphanumeric UIC: A format of user identification code (UIC) that specifies the user's group and member number in alphanumeric form rather than numeric form.

Attribute: In the security context, an attribute is a field of information maintained in the rights database that identifies some characteristic accorded to all holders of the identifier. For example, if an identifier possesses the resource attribute, holders of that identifier are able to charge resources such as disk space usage to that identifier.

Auditing: The act of noting the occurrence of an event that has security implications.

Authentication: The act of establishing the identity of users when they start to use the system. VMS (and most other commercial operating systems) use passwords as the primary authentication mechanism.

Breach: A break in the system security that results in admittance of a person or program to an object.

Break-In Attempt: An effort made by an unauthorized source to gain access to the system. Since the first system access is achieved through logging in, break-in attempts primarily refer to attempts to log in illegally. These attempts focus on supplying passwords for users known to have accounts on the system through informed guesses or other trial-and-error methods.

Captive Account: A type of VMS account that limits the activities of the user. Typically, the user is restricted to using certain command procedures and commands. The user may not be allowed to use the CTRL/Y key. (This type of account is synonymous with a turnkey or tied account.)

Decryption: The process that restores encoded information to its original unencoded form.

Discretionary Controls: Security controls that are applied at the user's option; that is, they are not required. Access control lists are typical of such optional security features. Discretionary controls are the oppposite of mandatory controls.

Disk Scavenging: A term that refers to any method of obtaining information from a disk that the owner intended to discard. The information, although no longer accessible to the original owner by normal means, retains a sufficient amount of its original magnetic encoding that it can be retrieved and used by one of the scavenging methods.

Encryption: A process of encoding information so that its content is no longer immediately obvious to anyone who obtains a copy of it.

Erasure Pattern: A character string that can be used to overwrite magnetic media for the purpose of erasing the information that was previously stored in that area.

Erase-on-allocate: A technique that applies an erasure pattern whenever a new area is allocated for a file's extent. The new area is erased with the erasure pattern so that subsequent attempts to read the area can only yield the erasure pattern and not some valuable remaining data. This technique is used to discourage disk scavenging.

Erase-on-delete: A technique that applies an erasure pattern whenever a file is deleted or purged. This technique is used to discourage disk scavenging.

Evasive Action: A responsive behavior by VMS to discourage break-in attempts when they appear to be in progress. VMS has a set of criteria it uses to detect the fact that break-in attempts may be underway. Typically, once VMS becomes suspicious that an unauthorized user is attempting to log in, the evasive action consists of locking out all login attempts by the offender for a limited period of time.

General Identifier: One of three possible types of identifiers that specify one or more groups of users. The general identifier is alphanumeric and typically is a convenient term that symbolizes the nature of the group of users. For example, typical general identifiers might be PAYROLL for all users allowed to run payroll applications or RESERVATIONS for operators at the reservations desk.

Highwater Marking: A technique for discouraging disk scavenging. This technique tracks the furthest extent that the owner of a file has written into the file's allocated area. It then prohibits any attempts at reading beyond the written area, on the premise that any information that exists beyond the currently written limit is information some user had intended to discard. VMS accomplishes the goals of highwater marking with its erase-on-allocate strategy.

Holder: A user who possesses a particular identifier. The term holder is used in conjunction with the term identifier. Users are said to be the holders of identifiers if they possess the identifiers. The rights database is the place in the system where the associations of users and the identifiers they hold are permanently kept. However, each process also has a right list that includes all the identifiers the process is authorized to hold.

Identifier: A notation that defines a user or group of users. There are three types of identifiers: UIC identifiers, system-defined identifiers, and general identifiers.

Locked Password: A password that cannot be changed by the account's owner. Only system managers or users with the SYSPRV privilege can change locked passwords.

Login: The series of actions involved in authenticating a user to the system and creating a process that runs on the user's behalf.

Mandatory Controls: Security controls that are imposed by the system upon all users. There are no examples of mandatory controls within the VMS system. Mandatory controls are the opposite of discretionary controls.

Nondiscretionary Controls: See *Mandatory Controls*

Open Accounts: Accounts that do not require passwords.

Passwords: Character strings that users provide at login time to validate their identity and as a form of proof of their authorization to access the account. There are system passwords and user passwords. User passwords include both primary and secondary passwords.

Primary Password: A type of user password that is the first user password requested from the user. Systems may optionally require a secondary password, as well. This password must be the password that is associated with the user name that is supplied with it.

Privileges: A means of protecting the use of certain system functions that can affect system resources and integrity. System managers grant privileges according to user's needs and deny them to users as a means of restricting their access to the system.

Proxy Login: A type of login that permits a user from a remote node to effectively log in to a local node as if the user owned an account on the local node. However, the user does not specify a password in the access control string. The remote user may own the account or share the account with other users.

Rights Database: The collection of data the system maintains and uses to associate identifiers and the holders of the identifiers with their rights and attributes.

Rights List: The list associated with each process that includes all the identifiers the process holds.

Secondary Password: A user password that may be required at login time immediately after the primary password has been correctly submitted. Primary and secondary passwords can be known by separate users to ensure that more than one user is present at the login. A less common use is to require a secondary password as a means of increasing the password length so that the total number of combinations of characters makes password guessing more time-consuming.

Secure Terminal Server: VMS software designed to ensure that users can only log in to terminals that are already logged out. When the user presses the BREAK key on a terminal, the secure server (if enabled) responds by first disconnecting any logged-in process and then initiating a login. If no process is logged in at the terminal, the login can proceed immediately.

Security Alarm: A message sent to operator terminals that are enabled as security operators. Security alarms are triggered by the occurrence of an event previously designated as worthy of the alarm because of its security implications.

Security Manager: In the VMS context, the person or persons responsible for protecting the security of the computer system. This role is sometimes performed by the same person who functions as a system manager. It requires the same skills as the system manager, but includes additional privilege (the SECURITY privilege) as well as knowledge of the security features provided with the VMS operating system.

Security Operator Terminal: A class of terminal that has been enabled to receive messages sent by OPCOM to "security operators." These messages are security alarm messages. Normally such a terminal is a hardcopy terminal in a protected room. The output provides a log of security-related events and details that identify the source of the event.

System-Defined Identifier: One of three classes of identifiers. System-defined identifiers are provided by the system to identify groups of users according to their usage of the system. For example, all users who access the system by dialing up receive the DIALUP identifier.

System Password: A password required by a terminal before login can be initiated.

Tied Account: See *Captive Account*

Trojan Horse Program: A program that gains access to otherwise secured areas through its pretext of serving one purpose when its real intent is far more devious and potentially damaging.

Turnkey Account: See *Captive Account*

UIC: See *User Identification Code*

User Identification Code: A coded notation that represents a user of the system. Normally each user has a unique user identification code, although at a few sites some users may share the same UIC. In that case, there is no way for the system to distinguish one user from another. User identification codes include a designation of the user and the user's group.

User Password: A password that is associated with a user. This password must be correctly supplied when the user attempts to log in so that the user is authenticated for access to the system. The two types of user passwords are known as primary and secondary, also the sequence in which they are entered.

Worm: A command procedure or executable image written and placed on the system for the sole purpose of seeking unauthorized access to files and accounts on the system. The "worm" seeks access to a user file through a flaw in the file protection. If successful, the worm modifies the file so that it carries a copy of the worm. Each time an unsuspecting user executes the code that contains the worm, the worm attempts to propagate itself into other poorly protected procedures or images, traveling along a path known as a worm-hole. The worm seeks to find its way into a procedure that will be run from a privileged account so that the worm can inflict damage to the system security.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Reader's Comments

Guide to VMS System Security
AA–LA40A–TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

_____

_____

What I like best about this manual is _____

_____

_____

What I like least about this manual is _____

_____

_____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

-- Do Not Tear - Fold Here and Tape ----------

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- Do Not Tear - Fold Here ----------

Cut Along Dotted Line

Reader's Comments

Guide to VMS System Security
AA–LA40A–TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

_____

_____

What I like best about this manual is _____

_____

_____

What I like least about this manual is _____

_____

_____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

-- Do Not Tear - Fold Here and Tape ----------

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- Do Not Tear - Fold Here ----------

Cut Along Dotted Line