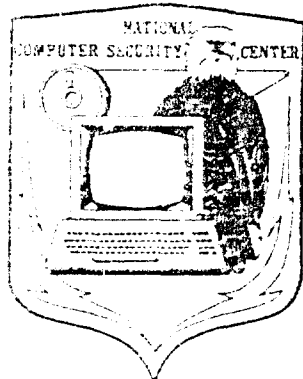




NATIONAL COMPUTER SECURITY CENTER

AD-A208 004

FINAL EVALUATION REPORT
OF
DIGITAL EQUIPMENT CORPORATION
VAX/VMS
VERSION 4.3

30 July 1986

DTIC
ELECTE
MAY 23 1989
S H D
cb

89 5 23 017

FINAL EVALUATION REPORT

DIGITAL EQUIPMENT CORPORATION

VAX/VMS VERSION 4.3

NATIONAL
COMPUTER SECURITY CENTER

9800 Savage Road
Fort George G. Meade
Maryland 20755-6000

July 30, 1986

CSC-EPL-86/004
Library No. S228,278

This page intentionally left blank.

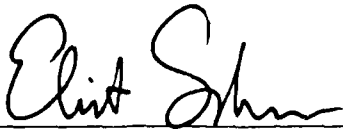
Final Evaluation Report Digital VAX/VMS Version 4.3

CSC-EPL-86/004
Library No. S228,278

FOREWORD

This publication, the Final Evaluation Report, Digital Equipment Corporation, VAX/VMS Version 4.3, is being issued by the National Computer Security Center under the authority of and in accordance with DoD Directive 5215.1, "Computer Security Evaluation Center". The purpose of this report is to document the results of the formal evaluation of Digital's VAX/VMS Version 4.3 operating system. The requirements stated in this report are taken from DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA, CSC-STD-001-83, dated 15 August 1983.

Approved:



Eliot Sohmer
Chief, Product Evaluations
and Technical Guidelines
National Computer Security Center



July 30, 1986

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

ACKNOWLEDGEMENTS

Team Members

Team members included the following individuals, who were provided by the following organizations:

Howard M. Israel
Shawn P. O'Brien

National Computer Security Center
Fort George G. Meade, Maryland

Jerzy W. Rub
Marilee J. Wheaton

The Aerospace Corporation
Los Angeles, California

Harriet G. Goldman

The MITRE Corporation
Bedford, Massachusetts

Further Acknowledgements

Acknowledgement is given to the following individuals for their contributions to this report: Linda Abraham, Stephen J. Covington, Henry L. Hall, Grace Hammonds, Jaisook Landauer, and Melissa W. Watson.

The following are trademarks of Digital Equipment Corporation: DEC, DECnet, DELNI, DEUNA, LA, MASSBUS, PDP, UNIBUS, VAX, VMS and VT.

MILVAX I and MILVAX II are trademarks of Norden Systems, Inc.

CONTENTS

	Page
Foreword	iii
Acknowledgements	iv
Executive Summary	ix
Section 1 Introduction	1
Evaluation Process Overview	1
Document Organization	2
Section 2 System Overview	3
Introduction	3
Processor State	5
User Process	6
Hardware Context	7
Software Context	7
Process Virtual Address Space	8
Memory Management	9
Mode Switching Instructions	10
Input/Output	10
Section 3 Evaluation as a C2 system	11
Discretionary Access Control	11
Identifiers	11
UIC Identifier	11
General Identifiers	12
System Identifiers	12
Access Control Lists	12
Identifier ACE	13
Default Protection ACE	14
Alarm ACE	14
UIC-Based Protection	15
Privileges	17
VAX/VMS Rules For Determining Access	17
Default Protection	18
Categorizing the Objects	18
Disk and Tape Volumes	19
Devices	19
Global Sections	19
Logical Name Tables	20
Queues	21
Mailboxes	21
Physical Memory	21
Object Reuse	21
Primary Memory	22
Disk Volumes	22
Disk Files	22

Final Evaluation Report Digital VAX/VMS Version 4.3

	Page
Erase-on-delete	22
Erase-on-allocate (Highwater marking)	23
Tape Volumes	23
Images and Global Sections	23
Identification and Authentication	24
Exceptions to User Authentication	25
Audit	25
System Architecture	28
Hardware Security Features	29
Processor Modes	29
Privileged Instructions	30
Memory	30
Software Security Features	31
TCB Layering	31
Processes	31
TCB Interface	32
System Integrity	33
Installation Testing	33
Error Logging	33
Hardware Diagnostics	34
Security Testing	34
Test Configuration	35
Security Features User's Guide	36
Trusted Facility Manual	37
Test Documentation	38
Identification and Authentication	39
Discretionary Access Control	39
Object Reuse	39
Audit	39
Design Documentation	40
Section 4 Additional Security Features	41
Trusted Path	41
Section 5 Evaluators' Comments	43
Discretionary Access Control	43
Identification and Authentication	43
Audit	44
Security Testing	44
Documentation	45
Miscellaneous	45
Appendix A Evaluated Hardware Components	A-1
Appendix B Evaluated Software Components	B-1
Appendix C VAX/VMS Privileges	C-1

Final Evaluation Report Digital VAX/VMS Version 4.3

Page

Appendix D	Acronyms	D-1
Appendix E	References	E-1

This page intentionally left blank.

Final Evaluation Report Digital VAX/VMS Version 4.3
Executive Summary

EXECUTIVE SUMMARY

The security protection provided by Digital Equipment Corporation's VAX/VMS Version 4.3 operating system (with "The VAX/VMS Systems Dispatch" article 95.5.8) has been evaluated by the National Computer Security Center (NCSC). The security features of VAX/VMS were evaluated against the requirements specified by the DoD Trusted Computer System Evaluation Criteria (the Criteria), CSC-STD-001-83, dated 15 August 1983.

The NCSC evaluation team has determined that the highest class at which VAX/VMS satisfies all the specified requirements of the Criteria is class C2. The evaluated hardware set includes the following stand-alone processors: 11/725, 11/730, 11/750, 11/751, 11/780, 11/782, 11/785, 8200, 8600, and 8650 as well as Norden Systems, Inc. MIL VAX I and II.

Class C2 systems provide a more finely grained discretionary access control than C1 systems, making users individually accountable for their actions through login procedures, auditing of security-relevant events, and resource isolation.

The VAX/VMS operating system is a general purpose time-sharing system with many security features. VAX/VMS provides hardware based protection through a hierarchical four mode protection scheme. This, in conjunction with its virtual memory protection capability, provides process separation. The Access Control List (ACL) mechanism provides discretionary access control.

This page intentionally left blank.

Final Evaluation Report Digital VAX/VMS Version 4.3
Introduction

INTRODUCTION

In September 1985, the National Computer Security Center (NCSC) began a formal product evaluation of VAX/VMS Version 4.2, a Digital Equipment Corporation product. The objective of this evaluation was to rate the VAX/VMS system against the Criteria and to place it on the Evaluated Products List (EPL). In April 1986, the NCSC and Digital mutually agreed to evaluate VAX/VMS Version 4.3, the most current release available. This report documents the results of the evaluation of VAX/VMS Version 4.3 released by Digital in December 1985.

Material for this report was gathered by the NCSC VAX/VMS evaluation team through documentation review, interaction with system developers, and experience using VAX/VMS systems.

Evaluation Process Overview

The Department of Defense Computer Security Center was established in January 1981 to encourage the widespread availability of trusted computer systems for use by facilities processing classified or other sensitive information. In August 1985 the name of the organization was changed to the National Computer Security Center. In order to assist in assessing the degree of trust one could place in a given computer system, the DoD Trusted Computer System Evaluation Criteria was written. The Criteria establishes specific requirements that a computer system must meet in order to achieve a predefined level of trustworthiness. The Criteria levels are arranged hierarchically into four major divisions of protection, each with certain security-relevant characteristics. These divisions are in turn subdivided into classes. To determine the division and class at which all requirements are met by a system, the system must be evaluated against the Criteria by an NCSC evaluation team.

The NCSC performs evaluations of computer products in varying stages of development from initial design to those that are commercially available. Product evaluations consist of a developmental phase and a formal phase. All evaluations begin with the developmental phase. The primary thrust of the developmental phase is an in-depth examination of a manufacturer's design for either a new trusted product or for security enhancements to an existing product. Since the developmental phase is based on design documentation and information supplied by the industry source, it involves no "hands on" use of the system. The developmental phase results in the production of an Initial Product Assessment Report (IPAR).

Final Evaluation Report Digital VAX/VMS Version 4.3

Introduction

The IPAR documents the evaluation team's understanding of the system based on the information presented by the vendor. Because the IPAR contains proprietary information, distribution is restricted to the vendor and the NCSC.

Products entering the formal phase must be complete security systems. In addition, the release being evaluated must not undergo any additional development. The formal phase is an analysis of the hardware and software components of a system, all system documentation, and a mapping of the security features and assurances to the Criteria. The analysis performed during the formal phase requires "hands on" testing (i.e., functional testing and, if applicable, penetration testing). The formal phase results in the production of a final report and an Evaluated Products List entry. The final report is a summary of the evaluation and includes the EPL rating which indicates the final class at which the product successfully met all Criteria requirements in terms of both features and assurances. The final report and EPL entry are made public.

Document Organization

This report consists of five major sections and five appendices. Section 1 is an introduction. Section 2 provides a general overview of the VAX system architecture. Section 3 provides a mapping between the requirements specified in the Criteria and the VAX/VMS features that fulfill those requirements. Section 4 describes an additional feature in the VAX/VMS system that satisfies a requirement at a class higher than the rated class. Section 5 provides an opportunity for the evaluation team to provide comments about the system.

Appendix A lists the evaluated hardware components. Appendix B lists the evaluated software components. Appendix C lists the privileges in VAX/VMS. Appendix D lists the acronyms and their expansions used in this report. Appendix E lists the references used in the writing of this report.

SYSTEM OVERVIEW

Introduction

VAX is a virtual address extension of the PDP-11 architecture. The system is distinguished by a 32-bit architecture, a memory management and protection mechanism, and a virtual memory operating system which provides a large program address space.

The VAX product line maintains a consistent architecture across processor lines and between microcode updates. The VAX hardware line consists of many different implementations of the VAX architecture.

The VAX architecture is defined by "Digital Equipment Corporation Standard 032" (see page E-1, "References"). This document defines the VAX architecture. It provides a complete description of the VAX central processor hardware as seen by machine language programs, and applies to all software written for the VAX processors, all VAX central processor hardware, and all closely-coupled VAX hardware peripherals. Thus, the consistent VAX architecture supports the portability of software across different hardware implementations.

The most important method of verifying that a new VAX system, or an update to an existing system, meets the architectural definition presented in "Digital Equipment Corporation Standard 032" is the use of a specialized tool. The tool's purpose is to verify the correct operation of the combination of instructions and operands defined by "Digital Equipment Corporation Standard 032". Conformance to this standard is tested by running test cases in each instruction group without any errors.

This architectural reference also describes the general methodology of performing virtual memory address translation and processor state transition, and defines a process' structure. The reference, however, does not identify any constraints or guidelines with respect to I/O. This implies that systems with vast differences in bus system speeds and bandwidths can still meet the architectural standard. The security-relevant aspect of I/O is that each implementation of the architecture (processor line) treats its own I/O subsystem in a uniform manner. Since this is the case among the VAX family of processors, the only remaining relevant issue is the bandwidth of the subsystem.

Final Evaluation Report Digital VAX/VMS Version 4.3 System Overview

However, bandwidth and covert channels are not a concern at the C2 level. (For a complete list of the evaluated hardware set, see page A-1, "Evaluated Hardware Components".)

The VAX family architecture is characterized by a set of 304 instructions. Those instructions that are specific to specialized portions of the VAX architecture (memory management, interrupts and exceptions, process dispatching, and process registers) are used by privileged software. In comparison with the PDP-11 architecture, the VAX architecture offers a wide range of data types, nine addressing modes, full demand paging memory management and a 4 gigabyte virtual address space.

VAX/VMS (Virtual Address eXtension/Virtual Memory System) is a multiuser, multifunction virtual memory operating system that supports multiple languages, an interactive command interface, and program development tools. The operating system performs process-oriented paging that allows execution of programs that may be larger than the physical memory allocated to them. Paging is handled automatically by the system.

The memory management facilities provided by VAX/VMS encompass both the memory protection and memory mapping mechanisms of VAX. Memory management software maintains tables of mapping information (page tables) that keep track of where each 512 byte virtual page is located in physical memory. The CPU uses this mapping information when it translates virtual addresses to physical addresses.

Memory management also provides page protection between processes since the process address of one process is not accessible to code running in the context of another process. When such accessibility is desired to share common routines or data, VMS provides controlled access through global sections (see page 19, "Global Sections").

In addition, four hierarchical access modes provide memory access control. Any location accessible to one mode is also accessible to higher privileged modes. (For an explanation of how the four access modes provide memory access protection see page 9, "Memory Management".)

A process can prevent pages from being paged out of its working set. With sufficient privilege, it can prevent its entire working set from being swapped out, to optimize program performance for real-time applications or interactive environments.

Final Evaluation Report Digital VAX/VMS Version 4.3 System Overview

VAX/VMS schedules CPU time and memory residency on a priority basis. Scheduling rotates among processes of the same priority. The scheduler adjusts the priority of processes assigned to one of the low 16 priorities to improve responsiveness and to overlap I/O and computation. Real-time processes can be placed in one of the top 16 scheduling priorities, in which case the scheduler does not alter their priority and does not preempt them. Therefore, real-time processes do not compete with lower priority processes for scheduling services. Their priorities can be adjusted by the system manager or an appropriately privileged user.

Data management is provided by VAX/VMS through a file and record management facility that allows the user to create, access, and maintain data files and records within files with full protection. The record management services (RMS) handles sequential, relative, and multikey indexed files. These files may have fixed or variable length records.

The VAX/VMS operating system supports the Files-11 On-Disk Structure Level 2 (ODS-2), which provides the facilities for file creation, extension, and deletion with owner-specified protections and hierarchical directories.

Processor State

The VAX family CPU provides sixteen 32-bit general registers which can be used for temporary storage, or as accumulators, index registers, and base registers. Registers R0 through R11 can be used as general purpose registers. The remaining four registers take on special significance depending on the instruction being executed: Register 12 (the Argument Pointer); Register 13 (the Frame Pointer); Register 14 (the Stack Pointer); and Register 15 (the Program Counter).

The CPU state consists of that portion of a process' state which, while the process is executing, is stored in processor registers rather than memory. The processor state includes the 16 general purpose registers, a 32-bit processor status longword (PSL), and privileged internal processor registers (IPR).

The process status longword (PSL) is a longword (32-bit) consisting of the privileged processor status concatenated with the processor status word (PSW). The PSW contains the condition codes that provide information on the results produced by previous instructions and the exception enables which control the processor action on certain exception conditions.

Final Evaluation Report Digital VAX/VMS Version 4.3 System Overview

The privileged internal processor register space provides access to different types of CPU control and status registers (e.g., the memory management base registers, parts of the PSL, and the multiple stack pointers). These registers are explicitly accessible only by the `Move_To_Processor_Register` (MTPR) and `Move_From_Processor_Register` (MFPR) instructions which require kernel mode (see page 9, "Memory Management") privilege.

User Process

An image is an executable program which is created by translating source language modules into object modules and then linking the object modules together. When a user initiates image execution, the image activator, part of the executive, sets up the process page tables to point to the appropriate sections of the image file. VAX/VMS uses the same paging mechanism that implements its virtual memory support to read image pages into physical memory as they are needed to execute it.

The environment in which an image executes is called its context. The complete context of an image includes not only the state of its execution at any one time (hardware context) but also the definition of its resource allocation privileges and resource quotas. Certain software information, including some key addresses and some software data structures, comprise the software context. An image executing in its context is called a process.

A process is the basic schedulable entity in the VAX/VMS system. A process consists of a virtual address space and hardware and software contexts. The hardware context of a process is defined by values that are loaded into processor registers when a process is scheduled for execution. When a process is not being executed, its hardware context is stored in the hardware process control block (hardware PCB). The hardware PCB is maintained by a software structure known as the process header (PHD). The act of saving the contents of the processor registers in the hardware PCB of the currently executing process and loading the new set of context from another hardware PCB is called context switching. The context switching operation uses two kernel instructions: the `Save_Process_Context` (SVPCTX) and `Load_Process_Context` (LDPCTX). The software context for each process is maintained in the software process control block (software PCB).

A job is defined to be the collection of subprocesses that has a common root process. The concept of a job exists solely for the purpose of sharing resources.

Hardware Context

The hardware context consists of copies of the general-purpose registers, the four per-process stack pointers, the program counter (PC), the processor status longword (PSL), and the process-specific processor registers including the memory management registers (see page 9, "Memory Management") and the asynchronous system trap (AST) level register implementing the software interrupt scheme. The hardware context is stored in the hardware process control block that is used primarily when a process is removed from or selected for execution.

There is one per-process stack for each of the four access modes. Any code that executes on behalf of a process uses one of that process' four stacks.

Software Context

The software context of a process consists of all the data required by the operating system to make scheduling and other control decisions about a process. This information includes the process priority, scheduling state, process privileges, resource quotas, process name and process ID.

The software PCB stores process data that must be constantly resident in memory. This includes the software priority of the process, its unique process identification (PID), the current process' scheduling state and some resource quotas. Those resource quotas that are shared among all processes in the same job are stored in the job information block (JIB). A JIB is a collection of images in the same process.

The process header data structure contains information about the process which does not have to be permanently memory resident (swappable process context). This information is required only when the process is resident and is used by memory management when a page fault occurs. Moreover, this data is used by the swapper when the process is removed from memory (outswapped) or brought into memory (inswapped). The hardware PCB, which contains the hardware context of the process, is also part of the process header. Whenever the process is resident, the data in the process header is available to appropriately privileged code.

Other process-specific information is stored in the control region of the process virtual address space (P1). This information is only accessible when the process is executing.

Final Evaluation Report Digital VAX/VMS Version 4.3 System Overview

Information stored in P1 space includes: exception dispatching information, RMS data tables, and information about the image currently executing.

Process Virtual Address Space

The basic addressable unit in VAX is the 8-bit byte. Virtual addresses are 32 bits long; resulting in the virtual address space of "2 to the 32" bytes. The virtual address space of a process is described by the process P0 and P1 page tables. These are stored in the high address end of the process header. A programmer uses a 32-bit virtual address to identify a byte location. This address is called a virtual address as it is not the real address of a physical memory location. It is translated into a real address by the processor under operating system control. A virtual address is not a unique address of a location in memory, as are physical memory addresses.

The set of all possible 32-bit virtual addresses is called virtual address space. This address space is equally divided into process space (P0, P1) and system space (S0, S1). A process can not represent virtual addresses in any process space but its own. Therefore, code and data belonging to a process are automatically protected from other processes in the system.

The process space is altered when an image is initially activated, during image execution through selected system services, and when an image terminates. The process page tables reside in system virtual address space and are described by entries in the the system page table. The page table entry (PTE) is the longword used to translate virtual addresses to physical ones. Unlike other portions of the process header, the process page tables are themselves pageable and are faulted into the process' working set only when they are needed.

Process space is divided into two equal regions: program region and control region. The addresses in the program region are used to identify the location of image code and data. Addresses in the control region are used to refer to stacks and other temporary program image and permanent process control information maintained by the operating system on behalf of the process.

Addresses in the remaining half of the virtual address space are used to refer to locations maintained and protected by the operating system. This space is referred to as system space. The operating system assigns specific meanings to addresses in system

Final Evaluation Report Digital VAX/VMS Version 4.3 System Overview

space; the same meaning for every process, independent of context. Locations referred to by system space addresses are protected from access by user images.

System space is also divided into two equal regions. Addresses in the first region (system region) are used for linkages to operating system service procedures, for memory management data, and for I/O processing routines. The second region of system space is currently unused.

Memory Management

The processor's memory management includes four hierarchical processor access modes that are used to provide read/write page protection between user software and system software. The access modes from most privileged to least are: kernel, executive, supervisor, and user. Kernel mode is used by the kernel for page management, scheduling, and I/O drivers. Executive mode is used for many of the operating system service calls, including RMS. Other services, such as command interpretation, are run in supervisor mode. Finally, user mode is used for user level code, utilities, compilers, debuggers, etc. Privileged data structures can be protected by not allowing read or write access to non-privileged users. A read or write permission at a less privileged mode is automatically granted at a more privileged mode.

Memory protection is the function of validating whether a particular type of memory access is allowed to a particular page. Access to each page is controlled by a protection code that specifies for each access mode whether or not read or write references are allowed. Additionally, each address is checked to make sure that it lies within the P0, P1, or system region. Attempts to access virtual addresses outside these bounds cause a length-violation fault. (For an explanation of how access mode switching occurs see page 10, "Mode Switching Instructions".)

Memory management registers are used to perform address translation, enable mapping, and to capture errors from memory management.

Every page in the virtual address space is protected according to its use. Even though all of the system space is shared, a program may be prevented from reading or modifying it. For example, in system space, scheduling queues are highly protected whereas library routines would be executable by code at any access mode.

Final Evaluation Report Digital VAX/VMS Version 4.3 System Overview

Mode Switching Instructions

There are four instructions that allow a program to change its access mode to the same or a more privileged mode and transfer control to a service dispatcher for the new mode using a standard CALL instruction. These are: Change_Mode_To_Kernel (CHMK), Change_Mode_To_Executive (CHME), Change_Mode_To_Supervisor (CHMS), and Change_Mode_To_User (CHMU).

These instructions provide the normal mechanism for less privileged code to call more privileged code. When the mode transition takes place, the previous mode is saved in the Previous Mode field of the PSL, thus allowing the more privileged code to determine the privilege of its caller.

In addition, there are two instructions which are used in conjunction with the above privileged instructions. The Probe_Read (PROBER) and Probe_Write (PROBEW) instructions allow privileged services to check addresses passed as parameters. The service routine always verifies that its less privileged caller could have directly referenced the address passed as parameters. Specifically, the PROBE instructions check the read and write accessibility of the first and last byte specified.

The operating system's privileged service procedures and interrupt and exception service routines exit using the Return_From_Exception_Or_Interrupt (REI) instruction. REI is the only way to decrease the privilege of the processor's access mode. The REI is used to restore the PC and the processor state to resume the process at the point where it was interrupted.

Input/Output

The I/O subsystem consists of device drivers and their associated data structures, device independent routines within the executive, and several system services. The most important system service is the \$QIO request, the I/O request that is issued by all outer layers of the system. I/O services perform input and output operations directly to a device driver, rather than through the file handling provided by RMS. The VAX architecture has no special instructions for input and output. Instead, inputs and outputs are performed by reads and writes of special physical memory addresses called CSRs (Control and Status Registers) and DBRs (Data Buffer Registers). Thus, the memory protection scheme also provides the protection of input and output.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

EVALUATION AS A C2 SYSTEM

Discretionary Access Control

Requirement

The TCB shall define and control access between named users and named objects (e.g., files and programs) in the ADP system. The enforcement mechanism (e.g., self/group/public controls, access control lists) shall allow users to specify and control sharing of those objects by named individuals, or defined groups of individuals, or both. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of including or excluding access to the granularity of a single user. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users.

Applicable Features

There are several components in VAX/VMS that provide discretionary access controls. These components are Access Control Lists, UIC (User Identification Code) based protection, privileges, and default protection mechanisms. The set of objects and their protection mechanisms that have been identified by the VAX/VMS evaluation team as being security-relevant will be discussed.

Identifiers

An identifier is a way to represent a user or a group of users in the system. There are three types of identifiers: UIC Identifiers, General Identifiers, and System Identifiers.

UIC Identifier

A UIC has two common formats: numeric and alphanumeric. A UIC in numeric representation contains a group number and member number. A UIC in alphanumeric format contains a member name and,

Final Evaluation Report Digital VAX/VMS Version 4.3 Evaluation as a C2 system

optionally, a group name. The member part of an alphanumeric UIC is equated to both the group and member parts of a numeric UIC in the system rights database. Therefore, member names must be unique. An example of a UIC would be [GROUP1, JONES], where JONES is a member of GROUP1.

General Identifiers

A General Identifier is a convenient way for the system manager to identify classes of users. They are created by adding the desired Identifier alphanumeric string to the Rights database, with the list of users that are to be associated with that General Identifier. An example might be BIOLOGY201 that contains a list of all students enrolled in that class.

System Identifiers

The System Identifiers are automatically created when the Rights database is created. System Identifiers describe particular types or classes of users. They are used to restrict the way in which a user can log into the system and can also be used in an Access Control Entry (ACE) to restrict access to an object based upon how the user has logged in to the system. VAX/VMS has six predefined System Identifiers: BATCH, NETWORK, INTERACTIVE, LOCAL, DIALUP, and REMOTE. An example of a System Identifier is BATCH. All Batch jobs carry the BATCH Identifier.

Access Control Lists

VAX/VMS supports Access Control Lists (ACLs) for five types of objects: files, directories, devices, global sections, and logical name tables. An ACL consists of a series of Access Control List Entries (ACEs). There are three different types of ACEs: Identifier ACEs, Default Protection ACEs, and Alarm ACEs.

The Identifier ACE controls the type of access allowed to specific users based on the user's identification; it specifies Read, Write, Execute, Delete, Control, or Null access rights to an object for the specified Identifiers.

The Default Protection ACE is only for directories. All files created under a directory with a Default Protection ACE will have their protection set to whatever is specified by the Default ACE. The effect of setting a default ACE occurs only on files created after the Default ACE has been created. Any files that existed before the Default ACE was created will not be affected.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

(However, the current protection could be changed on all existing files by using the SET PROTECTION command, so as to conform with the Default ACE.)

The Alarm ACE specifies whether auditing is to be performed and what type of accesses should be audited. Alarm ACEs can be created on a per object basis for files and directories. For auditing to occur, a privileged user must enable the alarm auditing mechanism (see page 25, "Audit").

An access control list may be added to the above listed five types of objects by use of a system service call (\$CHANGE_ACL), a DCL command (SET ACL/OBJECT=), or the ACL editor (EDIT/ACL). There are also commands for setting up default ACLs and displaying ACLs, as well as some helpful commands for duplicating ACLs (SET ACL/LIKE=). The search order for the ACL list is from first ACE to last stopping at the first match. The user is responsible for ordering ACEs correctly - usually from the most restrictive to the most general.

Identifier ACE

Identifier ACEs are used to define which access rights are allowed to Identifiers. An Identifier ACE specifies the access rights that are available to a specific UIC; the UIC must conform to the valid UIC format as described on page 11, "UIC Identifier".

An Identifier ACE can have one or more of the following options: DEFAULT, PROTECTED, NOPROPOGATE, and NONE. The options field controls whether an ACE is propagated or deleted.

The DEFAULT option is only valid for directories and is used to indicate that the ACE is to be included in the ACL of all files created within the directory - it is not used in determining directory protection. The PROTECTED option indicates that the ACE will be preserved even when an attempt is made to delete the ACL. The NOPROPOGATE option indicates that the ACE is not to be propagated to later versions of the file or from a parent directory to a newly created file in that directory. The NONE option specifies that there are no options for the ACE. If the NONE option is used in conjunction with any other options, the other options take precedence. The main purpose of the NONE option is for aesthetics and uniformity of the command language.

Final Evaluation Report Digital VAX/VMS Version 4.3 Evaluation as a C2 system

For each Identifier ACE the Access field of the ACE is used to define the access rights that are associated with the Identifier. The access rights can be set to one or more of the following: READ, WRITE, EXECUTE, DELETE, CONTROL, and NONE.

READ access allows all users associated with the Identifier to read a file, logical name table, global section, or disk, or allocate a device. WRITE access implies that the users can read or write. EXECUTE access allows users to execute an image file, DCL procedure file, or to look up entries in a directory without using the wildcard characters (special symbols which indicate "any" or "all" for a given field). DELETE allows for deletion of a file. CONTROL access grants the same privileges as the owner of the object and NONE grants a user no access to the object.

Default Protection ACE

The Default ACE is used to ensure that a particular UIC-based protection (a protection bit mask scheme; see page 15, "UIC-Based Protection") is propagated throughout a directory tree. Default protection ACEs can only be applied to directories. This ACE allows for the protection of a particular directory tree which is different than that applied to the default protections applied to directories in general. The protection mask field(s) of the Default ACE are the same as the UIC-based protection with the user categories: SYSTEM, OWNER, GROUP and WORLD, and the access categories: READ, WRITE, EXECUTE, and DELETE.

It is important to note that a user can not rely totally on directory protection to protect contained files. It is possible to open a file (by the use of file identification numbers rather than path names) which a user has access to even though the containing directory denies access (see "Guide to VAX/VMS System Security").

Alarm ACE

The Alarm ACE is used to specify that an alarm message is to be sent to the Security Administrator's terminal (or audit log) if the specified event takes place. An Alarm ACE applies only to disk files or directories and only functions when the Security Administrator has enabled auditing. The Alarm ACE can be set with one or more of the following options: DEFAULT, PROTECTED, NOPROPOGATE, and NONE.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

The DEFAULT option can only be applied to directories and is used to indicate that the Alarm ACE is to be included in the ACL of any file created within the directory. The PROTECTED option indicates that the Alarm ACE is to be preserved even when an attempt is made to delete the ACL. The NOPROPOGATE option is used to indicate that the Alarm ACE is not copied to later versions of the file. The NONE option indicates that no options are set for the Alarm ACE. If this option is used in conjunction with any other options, the other options take precedence (see page 25, "Audit").

UIC-Based Protection

UIC-based protection is determined by an owner UIC (see page 11, "UIC Identifier") and protection code. UIC-based protection mediates access to the following objects in the system: files, directories, volumes, devices, mailboxes, common event flag clusters, global sections, logical name tables, and queues. A volume is the medium which is logically mounted on a device. For example, disk packs and reels of magnetic tape are called volumes when they are mounted on disk and tape drives. For disk volumes, the system provides protection at the file, directory, and volume levels. For tape volumes, the system provides protection only at the volume level. For further information, see the "Guide to VAX/VMS Disk and Tape Operations".

When a user attempts to access a file or volume, the user's UIC is compared to the owner UIC of the object. Depending on the relationship of the UICs, the user falls into one or more of the following four categories (see "Guide to VAX/VMS System Security"):

1. SYSTEM

A user is granted the access rights specified in the system category of the UIC-based protection code associated with the object if one of the following is true:

- a. The user has system privilege (SYSPRV).
- b. The user has a group number between 1 and 10 (octal). These group numbers are generally for system managers, security managers, system programmers, operators and other privileged users. The range of numbers used can be changed by the system manager during SYSGEN. Users given these group numbers are implicitly given system privileges.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

c. The user has group privilege (GRPPRV) and the group field of the user's UIC matches the group field of the owner's UIC.

d. The user is the owner of the volume of the disk where the file is located. The owner UIC of the volume is set at the time the disk is initialized. This requires volume privilege (VOLPRO).

2. OWNER

A user is granted the access rights specified in the owner category of the UIC-based protection code associated with the object if the user's UIC matches the object's UIC.

3. GROUP

A user is granted the access rights specified in the group category of the UIC-based protection code associated with the object if the group field of the user's UIC matches the group field of object's UIC.

4. WORLD

Any user gains the access rights designated in the UIC-based protection code associated with the object.

Each of these categories of users can be granted or denied any of the following types of access: READ, WRITE, EXECUTE, and DELETE. A protection code describes the categories of users who have access to an object and the type of access that they have. For example, the file protection code:

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

specifies that users in the SYSTEM and OWNER categories have READ, WRITE, EXECUTE, and DELETE access. Users in the GROUP and WORLD categories have READ and EXECUTE access.

To determine access to a file, the system uses the resource's protection code for each user category. The system checks user categories from the outermost category (WORLD) to the innermost (SYSTEM). The user can access a resource as soon as the system

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

finds a user category that allows the user access. To deny access to a user category, all categories to the right of it must also be denied access. For example, even though the file protection code:

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

denies DELETE access to the file's owner, the owner can still delete the file because DELETE access is given under the WORLD category.

Privileges

VAX/VMS has thirty-five different privileges that can be associated with a process. The system administrator assigns privileges to a user via the AUTHORIZE utility. The thirty-five privileges are described in Appendix A (see page C-1, "VAX/VMS Privileges").

Four privileges directly affect the access a user actually receives regardless of the access dictated by an ACL or UIC-based protection mechanism. These four privileges are: SYSPRV (allows a user to access an object based on the SYSTEM UIC protection field), GRPPRV (allows access to an object based on the SYSTEM protection field when the process' group match the group of the object's owner), READALL (allows reading of an object despite any access protection), and BYPASS (allows a user to bypass all access protection on an object).

VAX/VMS Rules For Determining Access

VAX/VMS follows these steps to determine if a user is allowed access to a particular object:

1. VAX/VMS checks to see if an ACL exists. The ACL is searched from the first ACE to the last until the first match or the end. If the user is granted access in the ACL, then access is granted and further testing stops. If the user is not on the ACL, the system then uses UIC-based protection to determine access. If the user is refused access in the ACL, the system uses the SYSTEM and OWNER fields according to the UIC-based protection scheme to determine whether access should be granted. Access will be granted to the user if: 1) the user has privilege to do so; or, 2) the user is the owner of the object. The latter is to assure that the owner of an object is never denied access to the object.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

2. If the object does not have an associated ACL, the system uses UIC-based protection to determine access (see page 15, "UIC-Based Protection").

Default Protection

VAX/VMS has a series of default protection mechanisms. The system administrator can define a default UIC-based protection code for every volume that is mounted. In addition, the users can set default UIC-based protection codes or ACLs for their directories. When a default protection code has been established for a directory, all files and subdirectories created under that directory will have the default protection (see page 14, "Default Protection ACE").

Appendix C of "Guide to VAX/VMS System Security" lists the protections of all directories and files of VMS, as delivered by Digital.

Categorizing the Objects

VAX/VMS uses one or more of the DAC protection mechanisms described in the previous sections to protect objects and devices. This section describes the protection mechanisms that apply to specific objects.

Objects that can be used as communication paths (i.e., inter-process communication), such as Common Event Flags, Resource Locks, and Logical Names are objects that need not be controlled by the Trusted Computing Base (TCB) at the C Division. Logical names are the individual data elements contained in the logical name tables. Logical names could be used to establish a communication path between users. Since logical names are contained in the logical name tables, they are governed by the same access control mechanism outlined in the section on logical name tables (i.e., ACLs, UIC-based protection, and privileges).

The physical loading of a disk or tape to enable a user to mount the device, is handled by procedural methods. These methods are outlined in the "Guide to VAX/VMS System Security"

The following objects are internal to the TCB and are sufficiently protected by the TCB:

- Mailboxes
- Queues
- Physical Memory

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

The following objects are the security-relevant objects under control of the TCB:

- Devices
- Global Sections
- Files (ODS-2/Files-11)
- Directories
- Logical Name Tables
- Volumes

Disk and Tape Volumes

VAX/VMS supports the protection of data on disks at the volume and file level and magnetic tape media at the volume level only. The volume must be in Files-11 format. At the volume level, VAX/VMS provides UIC-based protection for both disks and magnetic tapes. At the file level, VAX/VMS provides discretionary access control through the use of ACLs for individual disk files and directory files that reside on disk volumes. In general, the protection that applies to a magnetic tape volume will also apply to all the files on that volume, (i.e., UIC-based protection). However, no protection is provided for non-Files-11 volumes. These volumes must be protected by procedural methods as described in the "Guide to VAX/VMS System Security".

An authorized user (one with VOLPRO privilege or whose UIC matches the volume UIC) may mount a Files-11 (a system-supported file structure) disk or tape volume with the /FOREIGN qualifier. Doing so gives the privileged user unlimited access to the contents of the volume, regardless of the ACL or UIC protection of files on the volume. Other users are prevented from accessing the volume during this time.

Devices

In addition to protecting the data on mounted volumes, VAX/VMS also provides device-level protection. For most non-file structured devices (e.g., terminals, card readers, line printers, laboratory devices, etc.), VAX/VMS provides ACL and UIC-based protection.

Global Sections

A Global section is a piece of code or data that is part of an image and can be shared among users. VAX/VMS provides both UIC-based protection and ACLs for Global Sections.

Logical Name Tables

VAX/VMS provides several logical name tables that are made available to the user. Logical name tables may be protected by ACLs and additional protection mechanisms described below.

1. LNM\$PROCESS_TABLE

The process table contains logical names that are available to the user's process. The system inserts logical names into this table at the time the user logs in. Users are allowed to modify their own process' logical name table.

2. LNM\$JOB_xxx

The job table (xxx is the unique number for the user's job tree) contains logical names that are available to the user's process and to any of its subprocesses. VAX/VMS inserts certain logical names in this table at the time the user logs in. Users can modify their own job table.

3. LNM\$GROUP_xxx

The group table (xxx is the GROUP number) contains logical names that are available to the Group. Only a user with GRPPRV or GRPNAM privilege can modify this table.

4. LNM\$SYSTEM

The system table contains logical names that are available to all users on the system. Only a user with SYSNAM or SYSPRV privilege can modify this table.

5. LNM\$PROCESS_DIRECTORY

The process directory is a logical name table containing the logical names of the logical name tables that are available to a process. This can be modified by the user.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

6. LNM\$SYSTEM_DIRECTORY

The system directory is a logical name table containing all the logical names of the logical name tables that are available to all users. Only a user with SYSNAM or SYSPRV privilege can modify this table.

Queues

VAX/VMS has two types of queue structures available: print and batch queues. To create or modify either of the queues requires OPER privilege. There are some utilities available in VAX/VMS that allow a user to see what jobs are waiting to be processed on the queues, however, the user can not modify any of the queue entries; the information contained in the queue entries is under the control of VAX/VMS. VAX/VMS provides UIC-based protection for queues.

Mailboxes

VAX/VMS mailboxes are used for interprocess communication and have both ACL and UIC-based protection.

Physical Memory

Memory is protected through the use of hardware protection mechanisms (see page 30, "Memory") and privileges.

Conclusion

VAX/VMS Version 4.3 satisfies the C2 Discretionary Access Control requirement.

Object Reuse

Requirement

When a storage object is initially assigned, allocated, or reallocated to a subject from the TCB's pool of unused storage objects, the TCB shall assure that the object contains no data for which the subject is not authorized.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

Applicable Features

VAX/VMS addresses object reuse with regard to primary memory, disk files and volumes.

Primary Memory

Newly allocated pages of primary (virtual) memory are always cleared or overwritten before being made accessible to a process. This procedure is enforced by VAX/VMS memory management and ensures that a user can not obtain residual information from the primary memory pages allocated to the user's process.

Disk Volumes

VAX/VMS allows a system administrator to clear a disk device of residual information before it is made available for use. This is accomplished when the volume is initialized with the ERASE attribute. This ensures that the volume starts out in a secure state. This same facility also enables erase-on-delete (see page 22, "Erase-on-delete") for the volume (see "Guide to VAX/VMS System Security").

Disk Files

Object reuse for disk files can be handled in either of two ways. One method involves the erasure of a disk file when it is deleted or purged (known as erase-on-delete). (A DELETE action removes the specified version(s) of the named file from the target directory. A PURGE action removes all versions, except the latest version, from the target directory.) The other method erases the file before it is allocated (known as erase-on-allocate or highwater marking). These two mechanisms are implemented in VAX/VMS to prevent disk scavenging.

Erase-on-delete

VAX/VMS enables disk files to be overwritten with a security erase pattern when deleted or purged. This feature is implemented voluntarily by the user with the ERASE qualifier on the DELETE and PURGE commands. The default security erase pattern is a pattern of zeros and is written once. The system administrator can, however, change the erasure pattern as well as the number of times the pattern is written (see "Guide to VAX/VMS System Security").

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

The erase-on-delete technique can also be implemented on an individual volume basis using the SET VOLUME/ERASE_ON_DELETE command. Once this command is executed, any file deleted or purged that resides on the volume is overwritten with the erasure pattern. The enforcement of erase-on-delete can be changed by the user only if the user has the necessary privilege (i.e., the user is owner of the volume, has a system UIC, or has SYSPRV privilege). Note: if a device has erase-on-delete or highwater marking specified, an unprivileged user can not override enforcement by the system by specifying the NOERASE attribute at the time of deletion or purging.

Erase-on-allocate (Highwater marking)

The other method by which VAX/VMS deals with disk file scavenging is through "highwater marking". Using the SET VOLUME/HIGHWATER_MARKING command ensures that all disk space to be allocated or extended to a process is erased before the process can access it. Highwater marking in VAX/VMS ensures that if a user attempts to read beyond the point in a file in which the user's process has written, the user will not obtain any information to which the user is not authorized (i.e., the space is overwritten with the security pattern as it is allocated).

Tape Volumes

No mechanism is provided to prevent a user from reading beyond the end of file on a magnetic tape. In addition, there is no enforceable mechanism for erasing a tape or a file on a tape. However, TU78 and MSCP tapes support erasure upon initialization. The ability to read residual information from magnetic tapes is not prevented in VAX/VMS. Object reuse for magnetic tapes must be handled by procedural methods or configured out of the system.

Images and Global Sections

Other objects that store data include images and global sections. Images and mapped global sections are stored in paged memory. Data within these objects are not subject to scavenging because the data in the related pages are erased when the pages are re-allocated.

Conclusion

VAX/VMS Version 4.3 satisfies the C2 Object Reuse requirement.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

Identification and Authentication

Requirement

The TCB shall require users to identify themselves to it before beginning to perform any other actions that the TCB is expected to mediate. Furthermore, the TCB shall use a protected mechanism (e.g., passwords) to authenticate the user's identity. The TCB shall protect authentication data so that it cannot be accessed by any unauthorized user. The TCB shall be able to enforce individual accountability by providing the capability to uniquely identify each individual ADP system user. The TCB shall also provide the capability of associating this identity with all auditable actions taken by that individual.

Applicable Features

VAX/VMS requires that all users be authenticated before any user processing is allowed, with exceptions noted below. The user must provide to the LOGINOUT.EXE process a username and password. The process then verifies the username and password against the User Authorization File (UAF) maintained by the System Administrator. In verifying the password, VMS first encrypts(1) the password entered by the user. The encrypted password is then compared to the value stored in the UAF. With the VAX/VMS encryption algorithm, a 64-bit hashed value is always created, regardless of the length of the password entered.

The UAF is protected by the UIC-based protection mechanism (see page 15, "UIC-Based Protection"), and can further be protected using an access control list.

Once a user is successfully logged in, the system associates the user's UIC with the newly created process. All auditable user actions are identified in terms of the UIC of the process that caused the action.

(1) The evaluation team did not examine the encryption algorithm or implementation.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

Exceptions to User Authentication

There are exceptions to the required user authentication. The following features should not be used in order to maintain individual accountability:

1. Open accounts which require no password. The user is not prompted for a password and can begin user processing immediately.
2. Automatic logins which permit users to login without specifying a username. VAX/VMS associates a username with the terminal. Passwords may be used, if enabled.

For a description of additional identification and authentication features provided by VAX/VMS see page 43, "Evaluators' Comments".

Conclusion

VAX/VMS Version 4.3 satisfies the C2 Identification and Authentication requirement.

Audit

Requirement

The TCB shall be able to create, maintain, and protect from modification or unauthorized access or destruction an audit trail of accesses to the objects it protects. The audit data shall be protected by the TCB so that read access to it is limited to those who are authorized for audit data. The TCB shall be able to record the following types of events: use of identification and authentication mechanisms, introduction of objects into a user's address space (e.g., file open, program initiation), deletion of objects, and actions taken by computer operators and system administrators and/or system security officers. The TCB shall also be able to audit any override of human-readable output markings. For each recorded event, the audit record shall identify: date and time of the event, user, type of event, and success or failure of the event. For identification/authentication events the origin of request (e.g., terminal ID) shall be included in the audit record. For events that introduce an object into

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

a user's address space and for object deletion events the audit record shall include the name of the object. The ADP system administrator shall be able to selectively audit the actions of any one or more users based on individual identity.

Applicable Features

VAX/VMS audits security-relevant events through a variety of logs including an operator log, an accounting log, and operator and system administrator session logs. Additionally, it is possible and may be desirable to enable the system console, or some other terminal, as a security console. This results in all security-relevant events normally logged in the operator log to also be displayed at this console. For auditing to occur, a privileged user must enable the auditing mechanism.

The auditable events include successful logins, unsuccessful login attempts, logouts, successful object accesses, object access failures, process initiations, process initiation failures, direct audit of certain privileges, privileged user actions, volume mounts and accesses to selected system files.

The ACCOUNTING utility is used to gather and report successful login information and login failures. ACCOUNTING can be used to selectively extract information from the accounting log file, SYS\$MANAGER:ACCOUNTING.DAT. The login failures are supplemented with extensive information as to the reason for failure, such as incorrect password, expired password, expired account, improper job class, improper day or time for login, or break-in prevention. The accounting log file also contains logout information and exit conditions. In addition, a security alarm feature can be used to record logins, login failures and breakin attempts. These events are recorded in the system's operator log, SYS\$MANAGER:OPERATOR.LOG. The operator log and the accounting log are, by default, accessible only by privileged users such as Security Administrators.

VAX/VMS performs complete auditing of mounting of user and system volumes. Recorded information includes the device type, label, logical name, owner, protection, requestor's name, and current date and time. The SET AUDIT command will cause the audit data to be placed in the operator log file and simultaneously displayed on the system console.

VAX/VMS provides the capability to audit selectable types of access on files and global sections. The files must be located on Files-11/ODS-2 structured disks. The selectable access types

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

are READ, WRITE, EXECUTE, DELETE, and CONTROL. These access types can be further defined by combining them with either successful accesses, failures, or both. For example, a security administrator may specify that only write failures be audited, or only read successes be audited.

Special user functions are finely controlled via a set of thirty-five privileges (see page C-1, "VAX/VMS Privileges"). However, VAX/VMS is only capable of directly auditing the use of four of these privileges. They are BYPASS, GRPPRV, READALL, and SYSPRV. File or global section accesses using these privileges can be audited. There are nine other security-relevant privileges which are not automatically audited. These nine privileges are CMEXEC and CMKRNL (both allow executing privileged instructions and full access), LOG_IO and PHY_IO (allows low-level I/O), PFNMAP (allows a user to map to specific physical pages in memory), SECURITY (allows a user to perform security-related functions), SETPRV (to turn on any privilege), SYSNAM (to modify system logical name table), and VOLPRO (which in conjunction with SYSNAM allows users to indirectly access any file on any volume).

However, use of all privileges can be indirectly audited. This indirect method involves creating executable files which perform desired privileged functions. Such files are first protected with proper ACLs and then installed in the operating system. Auditing accesses to these files will then produce a record of use of the privileges (see "Guide to VAX/VMS System Security").

VAX/VMS is capable of auditing successful and unsuccessful process creations, image activations and accesses to logical name tables.

All modification attempts on the user authorization file, SYS\$SYSTEM:SYSUAF.DAT, are also auditable, however, the audit entries include only the account name, the new parameter name and the user who performed the modification. (The actual value of the old parameter can only be audited via a session log.)

System Administrator and Operator actions can be recorded from a session log in a captive account. This will record all actions taken by the user.

In addition, VAX/VMS records the date and time of event, the user, and the event type. Success or failure of an action can also be audited. For identification and authentication events, the origin of the request is recorded and for object access events, the name of the object is specified.

Final Evaluation Report Digital VAX/VMS Version 4.3 Evaluation as a C2 system

VAX/VMS applies auditing criteria to a specified object or user. In addition, an audit reduction tool is provided with the system to allow further selectivity in extracting data from the audit logs. The Security Administrator may specify user names, particular time frames, or particular types of events as the audit reduction criteria.

Warnings concerning the possibility of audit log overflow will be generated by the system (see "The VAX/VMS Systems Dispatch", article 95.5.8). This will allow a System Administrator to take corrective action before audit records are lost due to a full system disk.

Conclusion

VAX/VMS Version 4.3 satisfies the C2 Audit requirement.

System Architecture

Requirement

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering (e.g., by modification of its code or data structures). Resources controlled by the TCB may be a defined subset of the subjects and objects in the ADP system. The TCB shall isolate the resources to be protected so that they are subject to the access control and auditing requirements.

Applicable Features

"Digital Equipment Corporation Standard 032" defines the VAX architecture (see page E-1, "References"). The security-relevant portions of the VAX architecture and of the VMS operating system are common on a wide range of hardware implementations. Norden Systems, Inc., which manufactures militarized implementations of the VAX architecture under license from Digital Equipment Corporation, also meet the standard and run VMS unchanged.

VAX/VMS provides several security-relevant protection mechanisms in hardware and software to provide process separation and resource isolation.

Hardware Security Features

Processor Modes

The VAX has four hierarchical processor protection modes (see page 10, "Mode Switching Instructions"). From most to least privileged they are: Kernel, Executive, Supervisor, and User. Each mode maintains its own stack. Movement from less privileged to the same or more privileged modes is performed by the execution of instructions or exceptions (software interrupts). These exceptions include the translation not valid (virtual address not mapped into physical memory) fault for memory management as well as the Change_Mode_To_Kernel, Change_Mode_To_Executive, Change_Mode_To_Supervisor, and Change_Mode_To_User instructions. These instructions are used by the system services which may execute in the same or a more privileged access mode (see page 10, "Mode Switching Instructions").

When a change mode instruction is executed, the processor switches to the stack of the specified mode. The Processor Status Longword (PSL) and Program Counter (PC) are saved on the new stack. The new PSL is formed with the caller's mode in the previous mode field and the current mode in the current mode field. The processor then runs in the higher mode. Because the change mode is really an interrupt, the address of the transfer is fixed by the system. Thus, it is possible for the system to do parameter checking and have a dispatch table for the final servicing. It is not possible for a user to jump to arbitrary locations at arbitrary processor modes. The instructions provide the only mechanism for less privileged code to call more privileged code.

When a mode transition occurs, the previous mode is save in the previous mode field of the PSL. This allows the more privileged code to determine the privilege of its most recent caller.

The Return_from_Exception_Or_Interrupt (REI) instruction provides a common return path and implements the protection scheme for the movement from more privileged to less privileged modes. When this instruction is executed, the top of the current stack contains the PSL and PC of the new mode. The processor verifies that the new mode is at the same or less privilege than the current mode. Thus, the instruction can not be used to increase the privilege of the processor access mode.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

Privileged Instructions

There are some instructions which may only be executed from Kernel mode. These are: HALT, Save_Process_Context (SVPCTX), Load_Process_Context (LDPCTX), Move_To_Processor_Register (MTPR), and Move_From_Processor_Register (MFPR). The SVPCTX and LDPCTX instructions assist VMS in the switching of process context by saving and restoring the general registers and memory management registers from the process control block (see page 6, "User Process"). The MTPR and MFPR provide access to processor internal privileged registers. Examples of these registers include: inner mode stack registers, base and bounds registers, interval counters, interrupt processors, and system identification.

Memory

The paged memory hierarchy is implemented so that a read or write to memory which is permissible in a less privileged mode will also be allowable in a more privileged mode. Memory protection is on a per-page (512 byte) basis. Privileged data structures can be protected by not allowing read nor write access to nonprivileged users.

The Probe_Read and Probe_Write instructions assist privileged mode routines in verifying that less privileged code actually has access to arguments which they are passing. It does this by checking the access protection of the memory page containing the argument as seen by the most recent caller.

Each process has its own virtual address space. Translation from virtual addresses to physical addresses is under strict control of VAX/VMS. The combination of processor modes, privileged instructions, and memory access protection provide VAX/VMS with its own domain and allows for process isolation by the use of distinct address spaces for user processes (see page 6, "User Process").

Software Security Features

TCB Layering

VMS is layered with the kernel executing in Kernel mode, the Record Management Services (RMS, a file system) in Executive mode, the Command Language Interpreter (CLI) in Supervisor mode, and user written images in User mode. Some system utilities installed with privileges execute in User mode. These images are protected by the use of special options in compilation and linking (to prevent the use of the interactive debugger) as well as control of the process address space.

Processes

The process is the fundamental subject of VMS. It is the entity selected for execution by the scheduler. A job is the collection of a process and all of its subprocesses. An image is the program a process executes in order to do meaningful work. A process is fully described by the hardware and software contexts and a virtual address space description.

The hardware context consists of the contexts of the general purpose registers, the four per-process stack pointers, the program counter, the processor status longword, the process specific processor registers including the memory management and asynchronous system trap (AST, an interrupt scheme) level registers. The software context includes information about priority, scheduling state, process privileges, resource quotas, and the unique process identification (PID).

Each process has its own user virtual address space (P0 process specific code and P1 process specific information) as well as shared system address space (S0). The process virtual address space is altered when an image is initially activated, during image execution through selected system services, and when an image terminates. The P0 and P1 address space of a process is not accessible to the code running in the context of another process.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

TCB Interface

The fundamental user interface with the TCB is the system services (see "VAX/VMS System Services Reference Manual"). The system services provide a uniform mechanism for the execution of system routines. These services are implemented by using the hardware protection schemes (mainly the change mode exception dispatch scheme discussed on page 29, "Processor Modes").

There are 110 system services which may be grouped into: security, event flags, AST, logical name, input/output, process control, timer and time conversion, condition handling, memory management, and lock management. One important group of system services is for security. This group includes services to implement the rights database, identifiers, access control list entries, access protection checking and security erase. These system services provide the primitive operations for many of the system utilities.

For example, users are defined by the system administrator by use of the AUTHORIZE utility. This utility makes use of the following system services: \$CREATE_RDB (create a rights database), \$ADD HOLDER (add holder record to rights database), \$ADD_IDENT (add identifier to rights database), \$FIND_HELD (returns identifiers held by a holder), \$FIND HOLDER (returns holders of an identifier in rights database), \$MOD HOLDER (modifies holder record in rights database), \$MOD_IDENT (modifies identifier record in rights database), \$REM HOLDER (deletes holder record from identifiers list of holders in rights database), \$REM_IDENT (deletes identifier and all holders of that identifier in rights database), and \$REVOKID (removes identifier from process or system rights list).

VMS isolates objects by allowing access only through system service calls, RMS calls, or CLI commands which all execute in more privileged processor modes than the user and which implement the security policy. The TCB provides isolation of its resources by controlling their manipulation through the system service interface.

Conclusion

VAX/VMS Version 4.3 satisfies the C2 System Architecture requirement.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

System Integrity

Requirement

Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TCB.

Applicable Features

VMS has a number of methods for verifying continued correct operation of the TCB.

Installation Testing

A utility called the User Environment Test Package (UETP) can be used to check system performance, both hardware and software. It is basically an exerciser that runs test loads that simulate users running programs which access disk files while checking file system inconsistencies. This permits checking bad quota, privilege, or account fields in the user authorization file (SYS\$SYSTEM:SYSUAF.DAT).

UETP also performs some minor diagnostics such as tape drive vacuum failure and device failure (either hardware or software) for devices that are connected to queues. UETP should only be regarded as a first phase of a comprehensive diagnostic procedure. The program does not try to diagnose the cause of the error but reports the error, quits that pass of the program, and continues with the next test. The error logger and Digital supplied diagnostics should be used to diagnose unknown problems.

Error Logging

The ERRFMT process keeps track of system errors, storing them in the file SYS\$ERRORLOG:ERRLOG.SYS. This file contains such hardware related errors as MASSBUS device errors, UNIBUS device errors, Error Correction Code (ECC) errors, Interrupt Exception Conditions (IEC), and machine checks, in addition to system software related errors such as bugchecks. Error correction codes repair single-bit errors and detect double-bit errors. This file also has an associated report generator, the Error Log utility, which is used to analyze this file.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

Another process with a similar function is the OPCOM process. The OPCOM process captures all operator output that is printed on the console as well as system events. This facility sends information that is used by operations personnel to the file SYS\$MANAGER:OPERATOR.LOG, such as security alarm messages, system volume mounts and dismounts (tape and disk), changes to the operating system parameters through SYSGEN, and user requests to mount and dismount volumes. There exists a utility (REPLY) which has as an option to close out the old OPERATOR.LOG file and open a new one. Both the currently used file and all previously used files are stored as ASCII text and thus can easily be reviewed.

Hardware Diagnostics

All VAX/VMS systems are supplied with a set of the standard VAX system hardware diagnostics. These diagnostics check the proper operation of the hardware, including testing the correct function of the microcode, and the data path circuitry used in the address translation logic of the virtual memory system.

Conclusion

VAX/VMS Version 4.3 satisfies the C2 System Integrity requirement.

Security Testing

Requirement

The security mechanisms of the ADP system shall be tested and found to work as claimed in the system documentation. Testing shall be done to assure that there are no obvious ways for an unauthorized user to bypass or otherwise defeat the security protection mechanisms of the TCB. Testing shall also include a search for obvious flaws that would allow violation of resource isolation, or that would permit unauthorized access to the audit or authentication data.

Applicable Features

The evaluation team performed functional testing of the security features of VAX/VMS Version 4.3 on a VAX 11/750, microcode revision level 98, in May 1986 at a Digital site. The security

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

features were tested, as per Section 10, "A Guideline on Security Testing" of the Criteria, and found to work as claimed in the system documentation. No obvious ways to bypass the security features of the system were discovered and no obvious flaws were found during testing or documentation review.

In addition to re-running the complete set of Digital's test suite (see page 38, "Test Documentation"), the team executed a set of 12 tests they designed. These tests were in the following general areas:

Auditing: verified that all auditable actions were recorded and the audit reduction tool worked as documented.

Discretionary Access Control: verified that the appropriate privilege was required before a privileged operation could take place.

Identification and Authentication: verified that various login restriction combinations and password restrictions were enforced.

Object Reuse: verified main memory was cleared and disk files spread over more than one volume were cleared.

Trusted Path: verified the functionality of the secure terminal server mechanism.

Test Configuration

The following is a description of the configuration that the evaluation team used to test VMS:

VAX 11/750 CPU with 6MB memory

2	-	RA60	205MB Disks
1	-	RL02	10.4MB Disk
1	-	TU77	Tape Drive
2	-	DZ11	8 Port Async MUX
1	-	DEUNA	Ethernet Controller
1	-	LA120	DecWriter III Console Terminal
1	-	VT100	Terminal
2	-	DF03	1200 Baud Modems

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

VAX 11/730 CPU with 3MB Memory

- 1 - R80 121 MB Disk
- 2 - RLO2 10.4 MB Disk
- 1 - TS11 Tape Drive
- 3 - DZ11 8 Port Async MUX
- 1 - LA120 DecWriter III Console Terminal
- 1 - DEUNA Ethernet Controller

- 1 - DELNI Local Network Interconnect
- 4 - VT100 Terminals
- 1 - LA100 Hardcopy Terminal

Conclusion

VAX/VMS Version 4.3 satisfies the C2 Security Testing requirement.

Security Features User's Guide

Requirement

A single summary, chapter, or manual in user documentation shall describe the protection mechanisms provided by the TCB, guidelines on their use, and how they interact with one another.

Applicable Features

Chapters 1-4 of the "Guide to VAX/VMS System Security" are written for the general user. These four chapters describe the protection mechanisms, guidelines on their use, and how they interact with one another.

Chapter 1 introduces the topic of security and levels of security requirements.

Chapter 2 introduces the reference monitor concept for security design and includes an overview of the security features provided by VMS.

Chapter 3 contains information on system security of interest to all users. This includes types of logins, information supplied at login time (e.g., last login message), the password facility, common causes of login failures and logging off.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

Chapter 4 describes the VMS object protection features that a user needs to know to protect the user's own information. This includes: UIC-based protection, access control lists, establishing and changing ownership of objects, default protection, preventing disk scanenging, user auditing, as well as a user self-test on protection techniques available.

Conclusion

VAX/VMS Version 4.3 satisfies the C2 Security Features User's Guide requirement.

Trusted Facility Manual

Requirement

A manual addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility. The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit event shall be given.

Applicable Features

Chapters 1-4 of the "Guide to VAX/VMS System Security" are written for the general user, and the other chapters in the guide are written for the system manager.

Chapter 5 describes VMS security features of particular interest to security managers. Specifically, section 5.8 describes the procedures for examining and maintaining the audit files. Additional audit information may be found in the "VAX/VMS DCL Dictionary", the "Guide to Networking on VAX/VMS", the "Guide to VAX/VMS System Management and Daily Operations" and in the "VAX/VMS Accounting Utility Reference Manual".

Chapter 6 provides guidance on recognizing when a system is under attack and summarizes the actions available to security managers to protect their systems. In addition, instructions on how to handle recent VMS security problems may be found in the "The VAX/VMS Systems Dispatch," published bimonthly. Chapter 7 describes security considerations for systems in networks. Chapter 8 describes security-related actions specific to clustered systems.

Final Evaluation Report Digital VAX/VMS Version 4.3 Evaluation as a C2 system

Appendix A summarizes all privileges available in VMS, the functionality they provide and which users should receive them. Appendix B describes how to modify user data in the User Authorization File. Appendix C lists the UIC-based protection that VMS provides, by default when delivered, for system files. Appendix D describes how to operate VMS in a C2 environment. Appendix E provides the detailed audit structure resulting from particular alarm events.

Conclusion

VAX/VMS Version 4.3 satisfies the C2 Trusted Facility Manual requirement.

Test Documentation

Requirement

The system developer shall provide to the evaluators a document that describes the test plan and results of the security mechanisms' functional testing.

Applicable Features

This requirement is satisfied by a test suite and the accompanying documentation that the evaluation team reviewed on-line in the 'test' directory of a Digital provided system. A majority of the the test suite were designed to execute in an automated manner.

Digital's test documentation suite uncovered two security flaws in an earlier version of VMS. These flaws were corrected in the evaluated version.

Digital also provided two documents that presented an overview to the test documentation suite:

- "VAX/VMS Security Evaluation Functional Test Plan"
- "VAX/VMS Security Evaluation Functional Testing Report For VAX/VMS Version 4.3".

The following is a brief description of the scope of tests described in the aforementioned reports.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

Identification and Authentication

The Login Control tests demonstrate the success and failure of user access attempts to the system. They verify that only those attempts where the user is properly identified to the system by means of an unique username and valid password will succeed.

The Process Rights Identifier tests demonstrate the use of the AUTHORIZE utility. This utility maintains the system authorization database used to create and modify user accounts on the system and to maintain a set of rights identifiers which are acquired by users when they gain access to the system.

Discretionary Access Control

For each class of objects protected by VMS the tests used a set of standard users possessing various combinations of UICs, system identifiers, general identifiers and privileges. Each of these users then attempted at least one Read, Write, Execute, Delete, and Control action against every object defined for testing that class.

Object Reuse

Three tests were developed that test the three different methods VMS uses to implement object reuse. The three methods are: 1) erase on delete (at the user's option), 2) erase on delete (site enforced), and 3) erase on allocate (highwater marking).

Audit

The audit tests showed that when an auditable event occurred, the proper audit record was written. Once all auditable events are created, the test uses the audit data reduction facility, SECAUDIT, to extract the audit information.

Conclusion

VAX/VMS Version 4.3 satisfies the C2 Test Documentation requirement.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluation as a C2 system

Design Documentation

Requirement

Documentation shall be available that provides a description of the manufacturer's philosophy of protection and an explanation of how this philosophy is translated into the TCB. If the TCB is composed of distinct modules, the interfaces between these modules shall be described.

Applicable Features

Chapter 2 of the "Guide to VAX/VMS System Security" documents the overall philosophy of protection underlying the design of VMS, and its translation into the specific features that are incorporated in the system.

The TCB is isolated into individual functional modules, whose interfaces are specified. This is documented in "VAX/VMS Internals and Data Structures".

Conclusion

VAX/VMS Version 4.3 satisfies the C2 Design Documentation requirement.

Final Evaluation Report Digital VAX/VMS Version 4.3
Additional Security Features

ADDITIONAL SECURITY FEATURES

Introduction

This section presents an evaluation of a VAX/VMS security feature that exceeds the C2 requirements. The feature described in this section does satisfy the higher level requirements of the Criteria in terms of the capability provided, but does not meet any of the assurance requirements above the C2 class.

Trusted Path

Requirement

The TCB shall support a trusted communication path between itself and users for initial login and authentication. Communications via this path shall be initiated exclusively by a user.

Applicable Features

There is a secure terminal server feature which invokes the login process on behalf of the user. This server can be invoked by the user pressing the Break key followed by a Return key on a terminal.

It can be implemented on a terminal by terminal basis. This facility will stop any currently executing process prior to the start of a login at that terminal. An application that uses the Break key for its own purpose will be affected by this option. The "Guide to VAX/VMS System Security" describes how to implement the feature on either direct connect or switched (e.g., modem) terminals.

Conclusion

VAX/VMS Version 4.3 satisfies(1) the B2 Trusted Path requirement.

(1) Although VAX/VMS Version 4.3 satisfies this requirement at the B2 level, it does not satisfy the assurance requirements above its rated level.

This page intentionally left blank.

EVALUATORS' COMMENTS

This section consists of comments and/or opinions from the evaluation team in the following areas:

- security features provided by the system (or vendor) that are not required by the Criteria,
- features provided by the system (or vendor) that are required by the Criteria but at a level higher than this evaluation was focused,
- the useability of some of the features that satisfy the Criteria requirements.

Discretionary Access Control

VAX/VMS contains many types of storage objects for many different purposes. Although there are protection mechanisms to permit sharing the different objects between users, the protection mechanisms are not consistent across all objects (e.g., there are ACLs and UIC-based protection for disk files, but only UIC protection for volumes).

Identification and Authentication

VMS also provides the capability to take 'evasive action' when a penetrator (possibly via an automatic password generator) has targeted a user's account. If a site-defined threshold of failed login attempts is exceeded, the account will be identified as one where a breakin is suspected. Real-time alarms may also be triggered. Once an account is under suspicion no logins can occur on that account, even if a valid password is entered. The threshold (site settable) is based upon the number of failed attempts per unit time. Only when the rate of attempted logins goes below the threshold can a valid login attempt occur on that account. This protection is also available on an account/port basis. This helps thwart a Denial of Service attack on system accounts/ports while still protecting the system.

VAX/VMS provides many useful tools for the Security Administrator to use for password management. This includes: the ability for VMS to force users to use system generated (pronounceable) passwords, password expiration, and minimum password length.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluators' Comments

Also, there is the capability for a site to specify a system-wide password to control access to particular terminals. The correct system password must be entered before the user receives the username prompt to begin logging in. For sensitive accounts, two separate passwords can be required before login is completed.

Audit

The system should have the capability to audit the use of privileges directly, via the audit mechanism, not just the assignment of them. Because of the ease of privilege proliferation, the proper management of privileges could go a long way towards increasing a site's security.

The system should have a capability to audit more system administrator actions directly (via the audit mechanism, instead of depending upon session logs) as well as modifications to access control lists by users.

The audit mechanism can suffer from useability problems.

The system should have the capability to audit the allocation of all devices.

It is worthy to note that the login information is not recorded in the accounting log (SYS\$MANAGER.ACCOUNTING.DAT) until a user logs out. However, user login and logout information can be obtained from the operator log (SYS\$MANAGER:OPERATOR.LOG).

Users have the capability to specify auditable events on files they own. These events are independent of what the System Administrator selects to be audited (however, the audit mechanism must be enabled by the System Administrator for any logging to occur).

VAX/VMS has the capability to generate real-time alarms on (user and/or System Administrator) selected events.

Security Testing

It is the team's view that the tests used to satisfy the Test Documentation requirement be incorporated into Digital's corporate quality assurance program.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluators' Comments

Documentation

"The VAX/VMS Guide to System Security" is good example of a compendium of information that a Security Administrator needs to know. It specifically addresses security issues, describes the security implications when selecting system wide options, recommends courses of action that an administrator could take in specific situations and provides many examples that highlight points made in the text.

Miscellaneous

There is a capability to limit user actions by removing commands that are available to users. This can be done on a user-by-user basis or on a system-wide basis.

There is a facility availability that permits an administrator the ability to limit space/time consumption (resources) on the machine. This will help protect against certain Denial of Service problems that can occur.

This page intentionally left blank.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Hardware Components

EVALUATED HARDWARE COMPONENTS

The evaluated hardware set includes the following stand-alone processors:

11/725

11/730

11/750

11/751

11/780

11/782

11/785

8200

8600

8650

Norden Systems, Inc. MIL VAX I

Norden Systems, Inc. MIL VAX II

This page intentionally left blank.

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

EVALUATED SOFTWARE COMPONENTS

The following TCB software listing was provided by Digital to the evaluation team:

This is a directory listing of a VMS V4.3 system disk, annotated to give the sensitivity and purpose of all files on the system disk.

There are seven levels of sensitivity, presented in rough order. The sensitivity number is interpreted as follows:

1. Primary TCB code.

Enforces protection policy / provides external interface to the TCB (e.g., exec, file system). Includes images installed with privileges susceptible to external attack.

2. Primary TCB data.

Contents is critical to the correct operation of the TCB (e.g., the authorization database).

3. TCB utility.

Used by the system administrator to manage the TCB. Does not directly enforce security policy in that its effective actions are mediated by other category (1) TCB components, but operates on TCB sensitive data (e.g., the authorization database), so operation must be correct to insure TCB integrity (e.g., AUTHORIZE utility).

4. Secondary TCB code.

Does not implement security policy or interface with users, but executes within the domain of the TCB and therefore must be correct to insure integrity of the TCB (e.g., device drivers). This category contains some images installed with privileges not readily subject to external attack (e.g., CMKRNL).

5. Secondary TCB data.

Data relevant to the TCB's operation, but not critical to its integrity. Contains confidential data (e.g., the accounting and audit logs).

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

6. Secondary TCB utility.

Utilities likely to be used by the system administrator in managing the system. Its effective actions are mediated by category (1) TCB components, and it does not operate on any security policy relevant structures, but incorrect operation could compromise the TCB (e.g., COPY command). Includes several shareable libraries used by category (1) and (4) components.

7. Not TCB.

Not involved in TCB operations, and not likely to be used in day to day management of the TCB. This does not mean that category (7) components are harmless. A trojan horse planted in any component used at some time by a system administrator can compromise the TCB. Directories have been given a rating based on the most sensitive object they contain.

The suffixes "N" and "C" are used on the sensitivity labels to indicate that the component is specific to networking or VAXclusters, respectively.

The bracketed name given after most .EXE files is the name of the facility under which the listings for that EXE may be found.

[000000]	1	Volume master file directory
000000.DIR	1	Volume master file directory
BACKUP.SYS	7	Not used by VMS
BADBLK.SYS	5	Volume bad block file
BADLOG.SYS	5	Pending bad block file
BITMAP.SYS	2	Volume storage allocation bitmap
CONTIN.SYS	7	Volume set continuation file
CORIMG.SYS	7	Not used by VMS
DECNET.DIR	7	User directory for the default DECnet account
INDEXF.SYS	2	Volume index file
SYS0.DIR	1	System top level directory
SYSEXE.DIR	4	Synonym system directory for booting
SYSMAINT.DIR	6	Field service working directory (synonym)
VOLSET.SYS	2	Volume set list
[SYS0]	1	System top level directory

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

SYS-CBI.DIR		7	Computer based instruction software
SYSERR.DIR		5	Error log files
SYSEXEC.DIR		1	Operating system executable images
SYSHELP.DIR		7	Help libraries
SYSLIB.DIR		1	Operating system libraries
SYSMAINT.DIR		6	Field service working directory
SYSMGR.DIR		2	System manager default directory
SYSMSG.DIR		7	System message files
SYSTEST.DIR		6	System regression tests
SYSUPD.DIR		3	System update tools
[SYS0.SYSERR]		5	Error log files
ERRLOG.SYS		5	System error log file
ERRSNAP.COM		6	VAX 8600 CPU snapshot copy facility
[SYS0.SYSEXEC]		1	Operating system files
ACC.EXE	[ACC]	3	Accounting log analysis utility
ACLEDIT.EXE	[ACLEDIT]	3	Access control list editor
ANALIMDMP.EXE	[IMGDMP]	4	Process dump analyzer (ANALYZE/PROCESS_DUMP)
ANALYZBAD.EXE	[BAD]	6	Bad block analysis utility (ANALYZE/MEDIA)
ANALYZOBJ.EXE	[ANALYZ]	7	Object file analysis utility (ANALYZE/ONJECT)
ANALYZRMS.EXE	[ANALYZ]	7	RMS record analysis utility (ANALYZE/RMS)
AUTHORIZE.EXE	[UAF]	3	Authorization database maintenance utility
AUTOGEN.PAR		2	SYSGEN parameter input to AUTOGEN
BACKUP.EXE	[BACKUP]	3	Backup/restore facility
BADBLOCK.EXE	[BADBLK]	4	Dynamic bad block analysis facility
BOOT58.EXE	[BOOTS]	4	TU58 conversational boot utility
BOOTBLOCK.EXE	[BOOTS]	4	Primary bootstrap for VAX-11/750
CDU.EXE	[CDU]	1	Command definition utility (SET COMMAND)
CHECKSUM.EXE	[UTIL32]	6	Software update checksum utility
CIA.EXE	[CLIUTL]	3	Intrusion list display (SHOW INTRUSION / DELETE/INTRUSION)
CLUSTERLOA.EXE	[SYSLOA]	4C	Cluster management kernel
CNDRIVER.EXE	[DRIVER]	4C	DECnet driver for the CI
CONFIGURE.EXE	[BOOTS]	4	MSCP controller configurator
CONINTERR.EXE	[DRIVER]	4	Connect to interrupt driver

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

CONVERT.EXE	[CONV]	7	RMS CONVERT utility
COPY.EXE	[COPY]	6	COPY utility
CRDRIVER.EXE	[DRIVER]	4	Card reader driver
CREATE.EXE	[CLIUTL]	6	CREATE utility (CREATE, CREATE/DIRECTORY)
CREATEFDL.EXE	[FDL]	7	CREATE/FDL utility
CSP.EXE	[SYSLOA]	4C	Cluster server process
CTDRIVER.EXE	[RTPAD]	4N	DECnet Cterm virtual terminal driver
CVDRIVER.EXE	[DRIVER]	4	VAX 8600 console RLO2 driver
CVTNAF.EXE	[UAF]	3	NETUAF V3/V4 conversion utility
CVTUAF.EXE	[UAF]	3	SYSUAF V3/V4 conversion utility
DBDRIVER.EXE	[DRIVER]	4	RP04/5/6 disk driver
DCL.EXE	[DCL]	1	DCL command language interpreter
DCLDEF.STB		7	DCL data structure definitions
DDDRIVER.EXE	[DRIVER]	4	TU58 driver
DELETE.EXE	[DELETE]	6	DELETE utility
DIFF.EXE	[DIF]	7	File compare utility
DIRECTORY.EXE	[DIR]	6	DIRECTORY utility
DISKQUOTA.EXE	[DISKQ]	3	Disk quota management utility
DISMOUNT.EXE	[DISMOU]	6	DISMOUNT command
DLDRIVER.EXE	[DRIVER]	4	RL01/2 disk driver
DMDRIVER.EXE	[DRIVER]	4	RK06/7 disk driver
DQDRIVER.EXE	[DRIVER]	4	VAX-11/730 integrated disk driver
DRDRIVER.EXE	[DRIVER]	4	RM03/5/RP07 disk driver
DSRINDEX.EXE	[RUNOFF]	7	RUNOFF index utility
DSRTOC.EXE	[RUNOFF]	7	RUNOFF table of contents utility
DTR.COM		7N	DECnet logical link test procedure
DTRECV.EXE	[DTS DTR]	7N	DECnet logical link test receiver
DTSEND.EXE	[DTS DTR]	7N	DECnet logical link test transmitter
DUDRIVER.EXE	[DRIVER]	4	DSA disk class driver
DUMP.EXE	[DUMP]	7	DUMP utility
DXDRIVER.EXE	[DRIVER]	4	Console RX01 driver
DYDRIVER.EXE	[DRIVER]	4	RX02 disk driver
DZDRIVER.EXE	[TTDRVR]	4	DZ11 terminal port driver
EDF.EXE	[EDF]	7	EDIT/FDL file description editor
EDT.EXE	[EDT]	6	EDT editor
ERF.EXE	[ERF]	6	Error log analysis utility (ANALYZE/ERROR)
ERFBRIEF.EXE	[ERF]	6	Error log analysis subcomponent
ERFBUS.EXE	[ERF]	6	Error log analysis subcomponent
ERFDISK.EXE	[ERF]	6	Error log analysis subcomponent
ERFINICOM.EXE	[ERF]	6	Error log analysis subcomponent
ERFPROC1.EXE	[ERF]	6	Error log analysis subcomponent

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

ERFPROC2.EXE	[ERF]	6	Error log analysis subcomponent
ERFPROC3.EXE	[ERF]	6	Error log analysis subcomponent
ERFPROC4.EXE	[ERF]	6	Error log analysis subcomponent
ERFPROC5.EXE	[ERF]	6	Error log analysis subcomponent
ERFRLTIM.EXE	[ERF]	6	Error log analysis subcomponent
ERFSUMM.EXE	[ERF]	6	Error log analysis subcomponent
ERFTAPE.EXE	[ERF]	6	Error log analysis subcomponent
ERFUVAX.EXE	[ERF]	6	Error log analysis subcomponent
ERFVX8600.EXE	[ERF]	6	Error log analysis subcomponent
ERRFMT.EXE	[ERRFMT]	4	Error log process
ERRSNAP.EXE	[ERRFMT]	6	VAX 8600 CPU snapshot copy utility
ESDRIVER.EXE	[DRIVER]	4	LANCE chip (integrated ethernet adaptor) port driver
EVL.COM		6N	DECnet event logger procedure
EVL.EXE	[EVL]	4N	DECnet event logger
EXCHANGE.EXE	[EXCHNG]	6	File exchange utility
F11AACP.EXE	[F11A]	1	Files-11 level 1 file system
F11BXQP.EXE	[F11X]	1	Files-11 level 2 file system
FAL.COM		6N	DECnet file server command file
FAL.EXE	[FAL]	6N	DECnet file server
FILESERV.EXE	[F11X]	4C	Cluster file system cache server
FPEMUL.EXE	[EMULAT]	4	Floating point emulator for microvax
FYDRIVER.EXE	[DUP]	4	DUP class driver (used by SET HOST/HSC)
HLD.COM		7N	RSX down-line task load procedure
HLD.EXE	[HLD]	7N	RSX down-line task load process
HSCPAD.EXE	[DUP]	6	HSC remote console utility (SET HOST/HSC)
IMGDEF.STB		7	Image header format definitions
INIT.EXE	[INIT]	1	Volume initialization utility
INPSMB.EXE	[INPSMB]	1	Batch job input symbiont
INSTALL.EXE	[INSTAL]	3	Installed file maintenance utility
JOBCTL.EXE	[JOBCTL]	1	Batch/print job controller
LADRIVER.EXE	[DRIVER]	4	LPA11 lab peripheral driver
LALOAD.EXE	[MCLDR]	6	LPA11 microcode load utility
LALoader.EXE	[MCLDR]	4	LPA11 microcode loader process
LATCP.EXE	[LAT]	6N	LAT control program
LCDRIVER.EXE	[DRIVER]	4	DMF32 line printer driver
LIBRARIAN.EXE	[LIBRAR]	6	Object/macro/help/text librarian utility
LINK.EXE	[LINKER]	6	Linker
LOGINOUT.EXE	[LOGIN]	1	LOGIN & LOGOUT utility
LPDRIVER.EXE	[DRIVER]	4	LP11 line printer driver
LTDRIVER.EXE	[LAT]	4N	LAT-11 terminal port driver
MACRO32.EXE	[MACRO]	7	Macro assembler

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

MAIL.COM		7	Mail receiver DECnet task
MAIL.EXE	[MAIL]	1	MAIL facility
MAILEDIT.COM		7	MAIL SEND/EDIT command file
MBXDRIVER.EXE	[DRIVER]	4	Shared memory mailbox driver
MESSAGE.EXE	[MSGFIL]	7	Message compiler utility
MIRROR.COM		7N	DECnet link loopback test procedure
MIRROR.EXE	[MIRROR]	7N	DECnet link loopback test process
MODPARAMS.DAT		2	SYSGEN parameters input to AUTOGEN
MOM.COM		6N	DECnet down line system load procedure
MOM.EXE	[MOM]	4N	DECnet down line system load process
MONITOR.EXE	[MONITOR]	1	MONITOR utility
MP.EXE	[MP]	4	VAX-11/782 multi-processor support
MP.STB		7	MP support symbol table
MSCP.EXE	[MSCP]	4C	MSCP server
MTAACCP.EXE	[MTAACCP]	1	Magtape file system
NCP.EXE	[NCP]	3N	Network control utility
NDDRIVER.EXE	[NETACP]	4N	DECnet down line load and loopback class driver
NETACP.EXE	[NETACP]	1N	DECnet protocol ACP
NETCIRC.DAT		2N	DECnet circuit database
NETCONF.DAT		2N	DECnet configurator database
NETDEF.STB		7N	Network structures symbol table
NETDRIVER.EXE	[NETACP]	4N	DECnet protocol driver
NETLINE.DAT		2N	DECnet line database
NETLOGING.DAT		2N	DECnet logging database
NETNODE.DAT		2N	DECnet node database
NETOBJECT.DAT		2N	DECnet object database
NETSERVER.COM		6N	DECnet general purpose listener procedure
NETSERVER.EXE	[NETACP]	6N	DECnet general purpose listener process
NETUAF.DAT		2N	DECnet proxy file (network authorization database)
NICONFIG.COM		6N	Ethernet module configurator procedure
NICONFIG.EXE	[NICNF]	4N	Ethernet module configurator process
NML.COM		6N	Network management listener procedure
NML.EXE	[NML]	6N	Network management listener process
NODRIVER.EXE	[DRIVER]	4N	DECnet asynch line class driver
NOTICE.TXT		7	Public notice file

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

OPCCRASH.EXE	[OPCOM]	6	Final system shutdown program
OPCOM.EXE	[OPCOM]	1	Operator communications manager (handles security audit log)
PADRIVER.EXE	[DRIVER]	4C	CI port driver
PAGEFILE.SYS		2	System paging file
PARAMS.DAT		2	System parameters input to AUTOGEN
PATCH.EXE	[PATCH]	6	Image patch facility
PDDRIVER.EXE	[DRIVER]	4	Pseudo-disk driver (used in TK50 boot)
PHONE.COM		7	PHONE listener task procedure
PHONE.EXE	[PHONE]	7	Phone utility
PRTSMB.EXE	[PRTSMB]	4	Print symbiont
PUDRIVER.EXE	[DRIVER]	4	Unibus port driver (UDA-50, TU81)
QUEMAN.EXE	[CLIUTL]	6	Queue management utility (SET / SHOW QUEUE)
RECLAIM.EXE	[CONV]	7	ISAM space reclamation utility (CONVERT/RECLAIM)
REMACP.EXE	[REM]	4N	DECnet virtual terminal protocol object
RENAME.EXE	[CLIUTL]	6	RENAME utility
REPLY.EXE	[OPCOM]	3	REPLY utility (REPLY/ENABLE /DISABLE)
REQUEST.EXE	[OPCOM]	7	REQUEST utility
RIGHTSLIST.DAT		2	Rights database
RMS.EXE	[RMS]	1	RMS - record management services
RMS.STB		7	RMS structure definitions
RMSDEF.STB		7	RMS structure definitions
RTB.EXE	[UTIL32]	6	Console bootstrap update utility
RTPAD.EXE	[RTPAD]	6N	DECnet virtual terminal utility (SET HOST)
RTTDRIVER.EXE	[DRIVER]	4N	DECnet virtual terminal driver
RUNDET.EXE	[CLIUTL]	6	Run process command utility
RUNOFF.EXE	[RUNOFF]	7	RUNOFF text formatter
RXDRIVER.EXE	[DRIVER]	4	Console RX50 disk driver
SCSDEF.STB		7C	SCS structure definitions
SCSLOA.EXE	[SYSLOA]	4C	CI system communication facility
SDA.EXE	[SDA]	6	Crash dump analyzer
SDLNPARSE.EXE		6	SDL pre-compiled language front end
SEARCH.EXE	[UTIL32]	7	SEARCH utility
SET.EXE	[CLIUTL]	1	SET commands
SETPO.EXE	[CLIUTL]	1	SET PASSWORD & SET MESSAGE commands
SETPARAMS.DAT		2	SYSGEN parameters input to AUTOGEN
SETSHOACL.EXE	[ACLEDT]	3	SET & SHOW ACL commands
SHOW.EXE	[CLIUTL]	4	SHOW commands

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

SHUTDOWN.COM		6	SHUTDOWN facility
SHWCLSTR.EXE	[CLIUTL]	1	SHOW CLUSTER utility
SMGBLDTRM.EXE	[SMGRTL]	6	Terminal characteristics table compiler
SMGMAPTRM.EXE	[SMGRTL]	6	Terminal characteristics table installation
SMGTERMS.TXT		7	Terminal characteristics table source
SORTMERGE.EXE	[SORT32]	7	SORT/MERGE utility
SRTTRN.EXE	[SORT32]	7	SORT/MERGE spec file translator
STABACCOP.EXE	[BACKUP]	6	Build stand-alone BACKUP kit
STABACKUP.EXE	[BACKUP]	3	Stand-alone BACKUP
STACONFIG.EXE	[BOOTS]	4	MSCP controller configurator during boot
STANDCONF.EXE	[BOOTS]	4	MSCP controller configurator for s/a BACKUP
STARTUP.COM		3	System startup command file
STARTUP.INS		3	Startup command file during installation
STASYSGEN.EXE	[BOOTS]	6	System configuration for stand-alone BACKUP
STOPREM.EXE	[REM]	6N	Utility to shut down REMACP
SUBMIT.EXE	[CLIUTL]	6	PRINT / SUBMIT commands
SUMSLP.EXE	[SUM]	6	EDIT/SUM utility
SWAPFILE.SYS		2	System swap file
SYS.EXE	[SYS]	1	System kernel
SYS.MAP		7	System link map
SYS.STB		6	System symbol table
SYSALF.DAT		2	Auto-login control file
SYSBOOT.EXE	[BOOTS]	4	Secondary system bootstrap
SYSDEF.STB		7	System data structure definitions
SYSDUMP.DMP		5	System crash dump file
SYSGEN.EXE	[BOOTS]	3	System configuration utility
SYSINIT.EXE	[SYSINI]	4	System initialization process
SYSLOA730.EXE	[SYSLOA]	4	CPU specific kernel code
SYSLOA750.EXE	[SYSLOA]	4	CPU specific kernel code
SYSLOA780.EXE	[SYSLOA]	4	CPU specific kernel code
SYSLOA790.EXE	[SYSLOA]	4	CPU specific kernel code
SYSLOA8SS.EXE	[SYSLOA]	4	CPU specific kernel code
SYSLOAUV1.EXE	[SYSLOA]	4	CPU specific kernel code
SYSLOAUV2.EXE	[SYSLOA]	4	CPU specific kernel code
SYSUAF.DAT		2	System authorization file
SYSUAF.RL2		6	Template system authorization file
TECO.EXE	[TECO]	7	TECO text editor
TERMTABLE.EXE	[SMGRTL]	7	Terminal characteristics table
TERMTABLE.TXT		7	User terminal char table source
TFDRIVER.EXE	[DRIVER]	4	TU78 magtape driver

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

TMDRIVER.EXE	[DRIVER]	4	TE16/TU77 magtape driver
TPU.EXE	[TPU]	6	TPU text editor
TSDRIVER.EXE	[DRIVER]	4	TS11/TU80 magtape driver
TTDRIVER.EXE	[TTDRVR]	4	Terminal class driver
TUDRIVER.EXE	[DRIVER]	4	DSA tape class driver
TYPE.EXE	[CLIUTL]	6	TYPE utility
UNLOCK.EXE	[CLIUTL]	7	UNLOCK command
VAXEMUL.EXE	[EMULAT]	4	Subsetted instruction emulator for microvax
VAXVMSSYS.PAR		2	Current SYSGEN parameters
VERIFY.EXE	[VERIFY]	3	File structure verification utility (ANALYZE/DISK)
VMB.EXE	[BOOTS]	4	Primary system bootstrap
VMOUNT.EXE	[MOUNT]	6	MOUNT command
VMSHELP.EXE	[HELP]	7	HELP facility
WRITEBOOT.EXE	[BOOTS]	6	Write primary boot block
XADRIVER.EXE	[DRIVER]	4	DR11-W general purpose interface driver
XDDRIVER.EXE	[DRIVER]	4N	DMP11/DMV11 multi-drop sync line driver
XEDRIVER.EXE	[DRIVER]	4N	Ethernet port driver
XFDRIVER.EXE	[DRIVER]	4	DR32 general purpose interface driver
XFLOADER.EXE	[MCLDR]	4	DR32 microcode loader
XGDRIVER.EXE	[DRIVER]	4N	DMF32 sync line driver
XMDRIVER.EXE	[DRIVER]	4N	DMC11/DMR11 sync line driver
XWDRIVER.EXE	[DRIVER]	4N	DUP11 bisync line driver
YCDRIVER.EXE	[TTDRVR]	4	DMF32 terminal port driver
YFDRIVER.EXE	[TTDRVR]	4	DHU11/DHV11 terminal port driver
[SYSO.SYSHLP]		7	Help libraries and on line documentation
ACLEDT.HLB			
ANLRMSHLP.HLB			
DEBUGHLP.HLB			
DISKQUOTA.HLB			
EDFHLP.HLB			
EDTHELP.HLB			
EDTVT100.DOC			
EDTVT52.DOC			
EXAMPLES.DIR			
EXCHNGHLP.HLB			
HELPLIB.HLB			
INSTALHLP.HLB			
LATCP.HLB			
MAILHELP.HLB			
MNRHELP.HLB			

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

NCPHELP.HLB
PATCHHELP.HLB
PHONEHELP.HLB
SDA.HLB
SHWCLHELP.HLB
SYSGEN.HLB
TECO.HLB
TPUHELP.HLB
UAFHELP.HLB
VMSTLRHLP.HLB

[SYSO.SYSHLP.EXAMPLES] 7 System coding examples

ADDRIVER.MAR
CONNECT.COM
DOD_ERAPAT.MAR
DRCOPY.PRM
DRCOPYBLD.COM
DRMAST.MAR
DRMASTER.FOR
DRSLAVE.FOR
DRSLV.MAR
DTE_DFO3.MAR
GBLSECUFO.MAR
LABCHNDEF.FOR
LABIO.OPT
LABIOACQ.FOR
LABIOCIN.MAR
LABIOCIN.OPT
LABIOCOM.FOR
LABIOCOMP.COM
LABIOCON.FOR
LABIOLINK.COM
LABIOPEAK.FOR
LABIOSAMP.FOR
LABIOSEC.FOR
LABIOSTAT.FOR
LABIOSTRT.COM
LABMBXDEF.FOR
LBRDEMO.COM
LBRDEMO.FOR
LBRMAC.MAR
LPATEST.FOR
LPMULT.B32
MAILCOMPRESS.COM
MAILCVT.COM
MAILUAF.COM
MONITOR.COM
MONSUM.COM

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

MSCPMOUNT.COM
PEAK.FOR
SCRFT.MAR
SUBMON.COM
SYSGTTSTR.MSG
TDRIVER.MAR
TESTLABIO.FOR
USSDISP.MAR
USSLNK.COM
USSTEST.MAR
USSTSTLNK.COM
XADRIVER.MAR
XALINK.MAR
XAMESSAGE.MAR
XATEST.COM
XATEST.FOR
XIDRIVER.MAR

[SYSO.SYSLIB]

1 System libraries

ACLEDIT.INI		6	ACL editor initialization file
ADARTL.EXE	[ADARTL]	7	ADA run time library
BASRTL.EXE	[BASRTL]	6	BASIC run time library
BASRTL2.EXE	[BASRTL]	7	BASIC run time library
CDDSHR.EXE	[CDD]	7	Common data dictionary stub module
CLIMAC.REQ		7	DCL call structure definitions
COBRTL.EXE	[COBRTL]	7	COBOL run time library
CONVSHR.EXE	[CONV]	6	Callable CONVERT facility
CRFSHR.EXE	[CRF]	6	Cross reference facility
DBGSSISHR.EXE	[DEBUG]	1	Debugger system service intercept
DCLTABLES.EXE	[CLD]	2	DCL command interpreter tables
DCXSHR.EXE	[DCX]	6	Data compression facility
DEBUG.EXE	[DEBUG]	7	Symbolic debugger
DEBUG.TPU		7	Debugger for TPU procedures
DELTA.EXE	[DELTA]	6	Privileged mode debugger
DELTA.OBJ		6	Linkable privileged mode debugger
DISMNTSHR.EXE	[DISMOU]	1	DISMOUNT facility
DTE_DFO3.EXE	[RTPAD]	7	DFO3 autodialer for SET HOST/DTE
DTE_DF112.EXE	[RTPAD]	7	DF112 autodialer for SET HOST/DTE
DYN SWITCH.EXE	[CLIUTL]	1N	Switch terminal line to async DECnet
EDTSECINI.GBL		6	EDT emulator TPU procedure (compiled)
EDTSECINI.TPU		7	EDT emulator TPU procedure (source)

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

EDTSHR.EXE	[EDT]	6	Callable EDT editor
ENCRYPshr.EXE	[ENCRYP]	6	Encryption facility stub module
ERFCOMMON.EXE	[ERF]	6	Error log analysis component
ERFCTLshr.EXE	[ERF]	6	Error log analysis component
ERFLIB.TLB		6	Error log supported device list
ERFSHR.EXE	[ERF]	6	Error log analysis component
ERFSHR2.EXE	[ERF]	6	Error log analysis component
EVESECINI.GBL		6	EVE emulator TPU procedure (compiled)
EVESECINI.TPU		7	EDT emulator TPU procedure (source)
FDLshr.EXE	[FDL]	6	Callable FDL utility
FORDEF.FOR		7	Fortran OTS message symbols
FORIOSDEF.FOR		7	Fortran OTS I/O message symbols
FORRTL.EXE	[FORRTL]	6	Fortran run time library
IMAGELIB.OLB		7	Shareable image library
IMGDMP.EXE	[IMGDMP]	4	Process dump facility
LBRshr.EXE	[LBR]	6	Callable librarian
LIB.MLB		7	System macro library
LIB.REQ		7	BLISS system macro library
LIBDEF.FOR		7	Fortran library symbol definitions
LIBRTL.EXE	[LIBRTL]	6	General run time library
LIBRTL2.EXE	[LIBRTL]	6	General run time library
MOUNTshr.EXE	[MOUNT]	1	MOUNT facility
MTHDEF.FOR		7	Fortran math symbol definitions
MTHRTL.EXE	[MTHRTL]	6	Common math library
NMLshr.EXE	[NML]	3N	Callable NML functions
PASRTL.EXE	[PASRTL]	6	PASCAL run time library
PLIRTL.EXE	[PLIRTL]	6	PL/I run time library
RPGRTL.EXE	[RPGRTL]	7	RPG run time library
SCNRTL.EXE	[SCNRTL]	7	SCAN language run time library
SCRshr.EXE	[VMSLIB]	6	Screen management package (old)
SECUREshr.EXE	[LOADSS]	1	Security system services
SIGDEF.FOR		7	Fortran condition symbol definitions
SMBSRVshr.EXE	[PRTSMB]	4	Symbiont shareable services
SMGshr.EXE	[SMGRTL]	6	Screen management package (new)
SORTshr.EXE	[SORT32]	6	Callable sort facility
STARLET.MLB		7	System macro library
STARLET.OLB		7	System object library
STARLET.REQ		7	System BLISS macro library
STARLETSD.TLB		7	System interface definitions - SDL intermediate
SUMshr.EXE	[SUM]	6	Callable EDIT/SUM editor
TPAMAC.REQ		7	TPARSE BLISS macro library
TPUSECINI.GBL		6	Default TPU initialization (compiled)
TPUshr.EXE	[TPU]	6	TPU callable text editor

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

TRACE.EXE	[TRACE]	7	Program taceback facility
VAXCCURSE.OLB		7	C language terminal support library
VAXCTRL.EXE	[CTRL]	7	C run time library
VAXCTRL.OLB		7	C linkable run time library
VAXCTRLG.OLB		7	C linkable run time library
VMSRTL.EXE	[VMSRTL]	7	Old run time library entry vectors
XFDEF.FOR		7	Fortran DR32 symbols
[SYSO.SYSMGR]		2	System manager working directory
ACCOUNTNG.DAT		5	Accounting log file
ALFMAINT.COM		3	Auto-login file (SYSALF) maintenance utility
EVL.LOG		5N	DECnet event log
LOADNET.COM		6N	DECnet process startup command file
LPAll1STRT.COM		6	LPAll startup command file
LTLOAD.COM		6N	LAT startup command file
MAKEROOT.COM		6C	Build system root on common cluster system disk
NETCONFIG.COM		6N	Configure network database
OPERATOR.LOG		5	Operator log file (contains security audit messages)
RTTLOAD.COM		6N	Load DECnet virtual terminal driver
SECAUDIT.COM		3	Analyze security audit log
STARTNET.COM		6N	Start up DECnet
SYCONFIG.COM		6	Site specific configuration command file
SYSHUTDWN.COM		6	Site specific shutdown file
SYSTARTUP.COM		3	Site specific startup command file
VMSIMAGES.COM		3	Install images during system startup
VMSIMAGES.DAT		2	List of installed images
[SYSO.SYSMSG]		7	Shareable message files
ADAMSG.EXE	[ADARTL]		
CLIUTLMSG.EXE	[MSGFIL]		
DBGTBKMSG.EXE	[MSGFIL]		
FILMNTMSG.EXE	[MSGFIL]		
NETWRKMSG.EXE	[MSGFIL]		
PASMSG.EXE	[PASRTL]		
PLIMSG.EXE	[PLIRTL]		
PRGDEVMSG.EXE	[MSGFIL]		
RPGMSG.EXE	[RPGRTL]		

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

SCNMSG.EXE	[SCNRTL]
SHRIMGMSG.EXE	[MSGFIL]
SYSMGTMSG.EXE	[MSGFIL]
SYMSG.EXE	[MSGFIL]
TPUMSG.EXE	[TPU]
VAXCMSG.EXE	[CRTL]

[SYSO.SYSTEST]

6 System test facility

TCNTRL.CLD	
UETCLIG00.COM	
UETCLIG00.DAT	
UETCLIG00.EXE	[UETPSY]
UETCOMS00.EXE	[UETP]
UETDISK00.EXE	[UETP]
UETDMPF00.EXE	[UETP]
UETDNET00.COM	
UETDNET00.DAT	
UETDR1W00.EXE	[UETP]
UETDR7800.EXE	[UETPSY]
UETFORT01.DAT	
UETFORT01.EXE	[UETP]
UETFORT02.EXE	[UETP]
UETFORT03.EXE	[UETP]
UETINIT00.EXE	[UETPSY]
UETINIT01.EXE	[UETPSY]
UETLOAD00.DAT	
UETLOAD02.COM	
UETLOAD03.COM	
UETLOAD04.COM	
UETLOAD05.COM	
UETLOAD06.COM	
UETLOAD07.COM	
UETLOAD08.COM	
UETLOAD09.COM	
UETLOAD10.COM	
UETLOAD11.COM	
UETLPAK00.EXE	[UETP]
UETMA7800.EXE	[UETPSY]
UETMEMY01.EXE	[UETPSY]
UETNETS00.EXE	[UETP]
UETP.COM	
UETPHAS00.EXE	[UETP]
UETRSXFOR.EXE	
UETSUPDEV.DAT	
UETTAPE00.COM	
UETTAPE00.EXE	[UETP]
UETTTY00.EXE	[UETP]
UETUNAS00.EXE	[UETP]

Final Evaluation Report Digital VAX/VMS Version 4.3
Evaluated Software Components

[SYSO.SYSUPD]	3	System update tools
AUTOGEN.COM	3	Generate SYSGEN parameters
BLISSREQ.TLR	6	System tailoring control file
BOOTBLDR.COM	6	Build 782 console floppies
BOOTUPD.COM	6	Update console boot medium
CONSCOPY.COM	6	Copy console boot medium
CVTNAF.COM	6N	Run network authorization file conversion
CVTUAF.COM	6	Run system authorization file conversion
DECNET.TLR	6	System tailoring control file
DEVELOP.TLR	6	System tailoring control file
DXCOPY.COM	6	Copy files between console medium and working directory
EXAMPLES.TLR	6	System tailoring control file
FILETOOLS.TLR	6	System tailoring control file
HELP.TLR	6	System tailoring control file
LIBDECOMP.COM	6	Decompress system libraries
LIBRARY.TLR	6	System tailoring control file
MANAGER.TLR	6	System tailoring control file
MISCTOOLS.TLR	6	System tailoring control file
QUEUES.TLR	6	System tailoring control file
REQUIRED.TLR	6	System tailoring control file
SETDEFBOO.COM	6	Set default boot command file
SPKITBLD.COM	6	Build software product kit
STABACKIT.COM	6	Build standalone BACKUP kit
SWAPFILES.COM	6	Adjust page, swap, and dump file sizes
TEXTTOOLS.TLR	6	System tailoring control file
UETP.TLR	6	System tailoring control file
VMSINSTAL.COM	6	Install VMS product updates
VMSKITBLD.COM	6	Build a VMS system disk
VMSKITBLD.DAT	6	Data file for VMSKITBLD.COM
VMSTAILOR.COM	6	Tailor limited environment VMS system
VMSUPDATE.COM	6	Install VMS product updates (obsolete)
[SYSEXE]	4	Top level dir for compatibility with old bootstraps
SYSBOOT.EXE	4	Secondary system bootstrap

This page intentionally left blank.

VAX/VMS PRIVILEGES

VMS has thirty-five privileges that can be associated with a process. The descriptions below were extracted from "Guide to VAX/VMS System Security", Appendix A.

1. [NO] ACNT

[Disallows]/allows the process to create processes for which no accounting messages are written.

2. [NO] ALLSPOOL

[Disallows]/allows the process to allocate spooled devices.

3. [NO] ALTPRI

[Disallows]/allows the process to set priority values.

4. [NO] BUGCHK

[Disallows]/allows the process to make bug check error log entries.

5. [NO] BYPASS

[Disallows]/allows the process to bypass ACL and UIC protection.

6. [NO] CMEXEC

[Disallows]/allows the process to change its mode to executive.

7. [NO] CMKRNL

[Disallows]/allows the process to change its mode to kernel.

8. [NO] DETACH

[Disallows]/allows the process to create detached processes.

9. [NO] DIAGNOSE

[Disallows]/allows the process to issue diagnostic I/O requests.

10. [NO] EXQUOTA

[Disallows]/allows the process to exceed its quota.

Final Evaluation Report Digital VAX/VMS Version 4.3
VAX/VMS Privileges

11. [NO] GROUP

[Disallows]/allows the process to control other processes in the same group.

12. [NO] GRPNAM

[Disallows]/allows the process to place names in the group logical name table.

13. [NO] GRPPRV

[Disallows]/allows the process to access files in its own group with all the access rights granted to the system category of user for those files.

14. [NO] LOG_IO

[Disallows]/allows the process to issue logical I/O requests to a device.

15. [NO] MOUNT

[Disallows]/allows the process to issue a mount volume QIO request.

16. [NO] NETMBX

[Disallows]/allows the process to create a network device.

17. [NO] OPER

[Disallows]/allows the process to perform operator functions.

18. [NO] PFNMAP

[Disallows]/allows the process to create or delete sections mapped by page frame number.

19. [NO] PHY_IO

[Disallows]/allows the process to issue physical I/O requests to a device.

20. [NO] PRMCEB

[Disallows]/allows the process to create permanent common event flag clusters.

Final Evaluation Report Digital VAX/VMS Version 4.3
VAX/VMS Privileges

21. [NO] PRMGBL

[Disallows]/allows the process to create permanent global sections.

22. [NO] PRMMBX

[Disallows]/allows the process to create permanent mailboxes.

23. [NO] PSWAPM

[Disallows]/allows the process to alter its swap mode.

24. [NO] READALL

[Disallows]/allows the process to bypass existing restrictions that would otherwise prevent the process from reading a file.

25. [NO] SECURITY

[Disallows]/allows the process to perform security-related functions such as enabling or disabling security audits or setting the system password.

26. [NO] SETPRV

[Disallows]/allows the process to create processes with higher privileges.

27. [NO] SHARE

[Disallows]/allows the process to assign a channel to a device, even if the channel is allocated to another process or subprocess.

28. [NO] SHMEM

[Disallows]/allows the process to create or delete data structures in shared memory.

29. [NO] SYSGBL

[Disallows]/allows the process to create system global sections.

30. [NO] SYSLOCK

[Disallows]/allows the process to request locks on system-wide resources.

Final Evaluation Report Digital VAX/VMS Version 4.3
VAX/VMS Privileges

31. [NO] SYSNAM

[Disallows]/allows the process to place names in the system logical name table.

32. [NO] SYSPRV

[Disallows]/allows access to files and other resources as if the user has a system UIC.

33. [NO] TMPMBX

[Disallows]/allows the process to create temporary mailboxes.

34. [NO] VOLPRO

[Disallows]/allows the process to override volume protection.

35. [NO] WORLD

[Disallows]/allows the process to control all other processes in the system.

ACRONYMS

ACE	Access Control List Entry
ACL	Access Control List
ADP	Automatic Data Processing
ASCII	American Standard Code for Information Interchange
AST	Asynchronous System Trap
CHME	Change_Mode_To_Executive instruction
CHMK	Change_Mode_To_Kernel instruction
CHMS	Change_Mode_To_Supervisor instruction
CHMU	Change_Mode_To_User instruction
CLI	Command Language Interpreter
CPU	Central Processing Unit
CSR	Control and Status Register
DBR	Data Buffer Register
DCL	Digital Command Language
DEC	Digital Equipment Corporation
ECC	Error Correction Code
EPL	Evaluated Products List
IEC	Interrupt Exception Conditions
IPAR	Initial Product Assessment Report
IPR	Internal Processor Register
JIB	Job Information Block
LDPCTX	Load_Process_Context instruction
MFPR	Move_From_Processor_Register instruction

Final Evaluation Report Digital VAX/VMS Version 4.3
Acronyms

MTPR	Move_To_Processor_Register instruction
NCSC	National Computer Security Center
ODS-2	On-Disk Structure Level 2
OPCOM	Operator Communication Manager
PO, P1	Process Space
PC	Program Counter
PCB	Process Control Block
PDP	Programmable Data Processor
PHD	Process Header
PID	Process Identification
PROBER	Probe_Read instruction
PROBEW	Probe_Write instruction
PSL	Processor Status Longword
PSW	Processor Status Word
PTE	Page Table Entry
REI	Return_From_Exception_Or_Interrupt instruction
RMS	Record Management Services
SO, S1	System Space
SVPCTX	Save_Process_Context instruction
TCB	Trusted Computing Base
UAF	User Authorization File
UETP	User Environment Test Package
UIC	User Identification Code
VAX	Virtual Address eXtension
VMS	Virtual Memory System

Final Evaluation Report Digital VAX/VMS Version 4.3
References

REFERENCES

"Digital Equipment Corporation Standard 032", A-DS-EL00032-0-0
Rev D, March 26, 1985. (PROPRIETARY)

Deitel, H. M., "An Introduction to Operating Systems",
Addison-Wesley, 1983.

"The DEC Dictionary: A Guide to Digital's Technical
Terminology", Digital Press, Burlington MA, 1984.

"Department of Defense Trusted Computer System Evaluation
Criteria", CSC-STD-001-83, 15 August 1983.

"Guide to Networking on VAX/VMS", AA-Y512A-TE, AA-Y512A-T1, July
1985, VAX/VMS Version 4.2, Digital Equipment Corporation,
Maynard, Massachusetts.

"Guide to Using DCL and Command Procedures on VAX/VMS",
AA-Y501A-TE, September 1984, VAX/VMS Version 4.0, Digital
Equipment Corporation, Maynard, Massachusetts.

"Guide to VAX/VMS Disk and Magnetic Tape Operations",
AA-Y506A-TE, September 1984, VAX/VMS Version 4.0, Digital
Equipment Corporation, Maynard, Massachusetts.

"Guide to VAX/VMS System Management and Daily Operations",
AA-Y507A-YE, September 1984, VAX/VMS Version 4.0, Digital
Equipment Corporation, Maynard, Massachusetts.

"Guide to VAX/VMS System Security", AA-Y510A-TE, AA-Y510A-T1,
July 1985, VAX/VMS Version 4.2, Digital Equipment Corporation,
Maynard, Massachusetts.

"VAX Architecture Handbook", Digital, 1981.

"VAX Hardware Handbook", Digital, 1982.

"VAX Hardware Handbook", Volume 1 - 1986, Digital, 1985.

"VAX Record Management Services Reference Manual", AA-Z503A-TE,
September, 1984, VAX/VMS Version 4.0, Digital Equipment
Corporation, Maynard, Massachusetts.

"VAX Software Handbook", Digital, 1982.

Final Evaluation Report Digital VAX/VMS Version 4.3
References

"VAX/VMS Access Control List Editor Reference Manual", AA-Z414A-TE, September 1984, VAX/VMS Version 4.0, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS Accounting Utility Reference Manual", AA-Z400A-TE, September 1984, VAX/VMS Version 4.0, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS Authorize Utility Reference Manual", AA-Z406A-TE, September 1984, VAX/VMS Version 4.0, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS DCL Dictionary", AA-Z200A-TE, AD-Z200A-T1, July 1985, VAX/VMS Version 4.2, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS Disk Quota Utility Reference Manual", AA-Z413A-TE, September 1984, Version 4.0, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS Error Log Utility Reference Manual", AA-Z402B-TE, July 1985, VAX/VMS Version 4.2, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS Glossary", AA-Z102A-TE, September 1984, VAX/VMS Version 4.0, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS Internals and Data Structures", Kenah, Lawrence J., and Bate, Simon F., EY-00014-DP, 1984, Digital Press.

"VAX/VMS I/O User's Reference Manual: Part I", AA-Z600A-TE, July 1985, VAX/VMS Version 4.2, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS I/O User's Reference Manual: Part II", AA-Z601A-TE, September 1984, VAX/VMS Version 4.0, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS Mount Utility Reference Manual", AA-Z424A-TE, September 1984, VAX/VMS Version 4.0, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS Release Notes", Version 4.3 AA-GY69A-TE, December 1985, VAX/VMS Version 4.3.

"VAX/VMS Security Evaluation Functional Test Plan", Revision 1, TR-GSG-86-001, Government Systems Group, Digital Equipment Corporation, Maynard, Massachusetts, 1 May 1986. (PROPRIETARY)

Final Evaluation Report Digital VAX/VMS Version 4.3
References

"VAX/VMS Security Evaluation Functional Testing Report For VAX/VMS Version 4.3" TR-GSG-86-002, Government Systems Group, Digital Equipment Corporation, Maynard, Massachusetts, 29 May 1986. (PROPRIETARY)

"The VAX/VMS Systems Dispatch", September 1986, AD-L034A-33, Digital Equipment Corporation, Maynard, Massachusetts.

"VAX/VMS System Services Reference Manual", AA-Z501B-TE, July 1985, VAX/VMS Version 4.2, Digital Equipment Corporation, Maynard, Massachusetts.

This page intentionally left blank.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS NONE		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE			DISTRIBUTION UNLIMITED		
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CSC-EPL-86/004			5. MONITORING ORGANIZATION REPORT NUMBER(S) S228,278		
6a. NAME OF PERFORMING ORGANIZATION National Computer Security Center		6b. OFFICE SYMBOL (If applicable) C12	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State, and ZIP Code) 9800 Savage Road Ft. George G. Meade, MD 20755-6000			7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code)			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) (U) Final Evaluation Report, Digital Equipment Corporation, VAX/VMS Version 4.3					
12. PERSONAL AUTHOR(S) Howard Israel, Shawn O'Brien, Jerzy Rub, Marilee Wheaton, Harriet Goldman					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Year, Month, Day) 860730	
15. PAGE COUNT 85					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP			
			Digital DEC; VAX/VMS; C2; EPL; NCSC		
			Trusted Computer System Evaluation Criteria; (C2)		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
The security protection provided by Digital Equipment Corporation's VAX/VMS Version 4.3 operating system has been evaluated by the National Computer Security Center (NCSC). The NCSC evaluation team has determined that the highest class at which VAX/VMS satisfies all the specified requirements of the <u>Trusted Computer System Evaluation Criteria</u> is class C2, and therefore has been assigned a class C2 rating by the NCSC. A system that has been evaluated as being a class C2 system provides a Trusted Computing Base (TCB) that enforces discretionary access control and, through the use of audit capabilities, accountability for the actions users initiate. The VAX/VMS operating system is a general purpose time-sharing system with many security features. VAX/VMS provides hardware based protection through a hierarchical four mode protection scheme. This, in conjunction with its virtual memory protection capability, provides process separation. The Access Control List (ACL) mechanism provides discretionary access control.					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL LTC Lloyd D. Gary, USA			22b. TELEPHONE (Include Area Code) (301)859-4458		22c. OFFICE SYMBOL C/C12

UNCLASSIFIED