

To the Graduate Council:

I am submitting herewith a thesis written by Susan E. Hicks entitled "A Transaction Flow Approach To A Computer Operating Environment Security Evaluation." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

William J. McClain
William J. McClain, Major Professor

We have read this thesis
and recommend its acceptance:

JJS =

Tom Duray

Accepted for the Council:

Cewminkel
Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Susan E. Hicks

Date 11-13-89

A TRANSACTION FLOW APPROACH TO A
COMPUTER OPERATING ENVIRONMENT
SECURITY EVALUATION

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Susan E. Hicks
December 1989

Acknowledgements

I would like to acknowledge my appreciation to Dr. William J. McClain for his guidance as my thesis advisor. I am also grateful to Dr. Tom Dunigan and Dr. David Straight for their cooperation and assistance as members of my thesis committee.

I would like to acknowledge Martin Marietta Energy Systems corporate support of my academic efforts and the professional support of my colleagues.

A very special appreciation is extended to Dr. Rowena Chester of the Data Systems Research and Development Division of Martin Marietta Energy Systems for her support and encouragement in my thesis effort.

Abstract

This thesis presents the application of a software security evaluation methodology to a computer operating environment based on the VAX/VMS operating system. The evaluation methodology is based on a transaction flow analysis approach that involves the systematic analysis of security exposures and controls in a computer operating environment. The system is viewed as a set of transactions performed on the objects of the system with each transaction represented in a data flow diagram. The objectives of the evaluation methodology are to identify the exposures that exist in a system, to determine the controls provided to protect those exposures, and to assess the effectiveness of the controls in preventing or detecting exploitation of the exposures.

CONTENTS

1. INTRODUCTION	1
1.1 BACKGROUND	2
1.2 PURPOSE	2
1.3 APPROACH	4
2. EVALUATION APPROACH	6
2.1 TRANSACTION FLOW ANALYSIS	6
2.2 SECURITY REQUIREMENTS	8
3. BASIC EVALUATION	9
3.1 BASIC EVALUATION METHODOLOGY	9
3.2 OUTLINE OF STEPS REQUIRED TO PERFORM THE BASIC EVALUATION	9
3.3 EVALUATING THE SYSTEM SECURITY REQUIREMENTS . . .	11
3.4 AGENCY REQUIREMENTS	12
3.4.1 Security Policy	12
3.4.1.1 Discretionary Access Controls	12
3.4.1.2 Object Reuse	13
3.4.1.3 Labels	13
3.4.1.4 Mandatory Access Controls	14
3.4.2 Accountability	15
3.4.2.1 Identification and Authentication	15

3.4.2.2 Audit	16
3.4.3 Assurance	17
3.4.3.1 Operational Assurance	18
3.4.3.2 Life-Cycle Assurance	20
3.4.4 Documentation	21
3.4.4.1 Security Features User's Guide	21
3.4.4.2 System Security Administrator's Manual	21
3.4.4.3 Test Documentation	22
3.4.4.4 Design Documentation	22
3.4.5 Administrative Requirements	22
3.5 IDENTIFYING THE EXPOSURES	24
3.6 IDENTIFYING THE CONTROLS	24
3.7 COMPUTER OPERATING ENVIRONMENT CONTROLS	24
3.7.1 VMS System Access Controls	25
3.7.2 VMS Object Access Controls	29
3.7.2.1 UIC-based Protection	30
3.7.2.2 ACL Protection	31
3.7.2.3 Attributes and Privileges	32
3.7.2.4 Restricting User Capabilities	34
3.7.3 File and Directory Ownership	34
3.7.4 Volume Ownership	35

3.7.5 VMS Protection Defaults	35
3.7.6 Reuse	36
3.7.7 Operational Security	37
3.7.8 Auditing	38
3.7.9 Documentation and Life Cycle Controls	40
3.7.10 Local Site Controls	40
3.7.11 Agency Administrative Controls	40
3.8 MAPPING EXPOSURES AND CONTROLS	41
3.9 DETERMINING SYSTEM OBJECTS	42
3.10 DETERMINING TRANSACTION FLOWS	42
3.11 EVALUATION OF TRANSACTION FLOWS FOR EXPOSURES	51
3.12 EVALUATING CONTROL ADEQUACY	51
3.13 DATABASE	54
3.14 BASIC EVALUATION RESULTS	58
4. DETAILED EVALUATION	59
4.1 DETAILED EVALUATION METHODOLOGY	59
4.2 OUTLINE OF STEPS REQUIRED TO PERFORM THE DETAILED EVALUATION	59
4.3 IDENTIFYING TEST SCRIPTS	61
4.4 PERFORMING TESTING ACTIVITIES	61
4.5 REPORTING TEST RESULTS	61

4.6 DETAILED EVALUATION RESULTS	62
5. CONCLUSIONS	64
5.1 OVERALL EVALUATION RESULTS	64
5.2 METHODOLOGY EFFECTIVENESS	64
5.3 RECOMMENDATIONS	64
BIBLIOGRAPHY	66
APPENDIXES	70
APPENDIX A: AGENCY REQUIREMENTS AND EXPOSURES	71
APPENDIX B: AGENCY CONTROLS	86
APPENDIX C: EXPOSURES AND CONTROLS	96
APPENDIX D: TRANSACTION TEST SCRIPTS	102
VITA	114

LIST OF FIGURES

<u>Fig.</u>		<u>Page</u>
1	The Evaluation process	7
2	The Basic Evaluation process	10
3	Transaction Flow Diagram for the READ function	44
4	Transaction Flow Diagram for the WRITE function	45
5	Transaction Flow Diagram for the EXECUTE function	46
6	Transaction Flow Diagram for the DELETE function	47
7	Transaction Flow Diagram for the LOGIN Subsystem	48
8	Transaction Flow Diagram for the ACCESS CONTROL Subsystem	50
9	The database relationships	57
10	The Detailed Evaluation process	60

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	VMS logins.	25
2	Control Effectiveness	52
3	Database tables and elements.	54
4	Links between database tables.	54

1. INTRODUCTION

Computer security has become a growing concern with the increasing dependence on computer systems to perform the basic needs of our society. Computer systems are used in the business, academic, and government communities to process massive amounts of data that are considered to be of a sensitive nature. This sensitive data could include personal information about an individual, proprietary information, or information concerning national defense.

The use of multi-user computer systems has resulted in a need for controls to mediate access to the information processed on those computer systems, as well as access to the system resources shared among the users. Most modern multi-user operating systems provide control measures to mediate the access to resources that are shared among users and to protect system and user processes from unauthorized access by other user processes. Also, access control mechanisms are usually provided to protect stored data from access by unauthorized users. However, many of these security features usually provided by the operating system are optional and must be enabled to serve as effective control measures.

The National Computer Security Center (NCSC) has established evaluation criteria by which computer systems can be rated for security. The NCSC performs a formal verification and validation process to determine the security rating of a computer system. The criteria are published in the *Department of Defense Trusted Computer System Evaluation Criteria*¹, more commonly known as *The Orange Book*. These evaluation criteria are the bases for most computer system security certifications.

As computer security has become more of an issue, several terms have evolved to reference the items of concern. An *exposure* is a form of possible harm in a computing system, such as exposure of sensitive data to unauthorized users, or the

potential for modification or deletion of data, and denial of services to an authorized user. A *control* is a protective measure that reduces or prevents the exploitation of an exposure. A control could be an action, a device, a procedure, or a technique⁴¹. The data or system resources that are in need of protective measures are referred to as *objects*; *subjects* are the users or processes that access objects.

This thesis presents a methodology to evaluate the security exposures that exist in a computer operating environment and the controls that are implemented to prevent or to reduce exploitation of these exposures.

1.1 BACKGROUND

The work presented here considers the computer operating environments of a particular government agency that controls the computer operations of several individual sites. Although all of these sites are required to comply to a basic set of orders for the operation of computer systems, some sites have implemented additional security controls that are specific to the particular site.

The primary operating system used in this agency's computer operating environments is VAX/VMS*. Thus, the evaluation approach was applied to an operating environment based on this operating system to expedite development of the testing methodology. The security evaluation discussed in this thesis concerns only stand-alone systems. Although the methodology could be applied to networked systems or to operating environments using other operating systems, those systems are beyond the scope of this study.

1.2 PURPOSE

The government agency for which this work was performed requested the development of a system security evaluation methodology that could be applied to all of its operating sites in a consistent manner. The methodology was to be presented in the form of a generic test plan²⁵ that could be used by local security

* VAX/VMS is a product of Digital Equipment Corporation.

evaluators or auditors to perform security evaluations of the local site computer systems.

Although this government agency is not a Department of Defense (DOD) agency, most of the agency's current computer systems comply to the requirements for a C2 level system as defined in *The Department of Defense Trusted Computer System Evaluation Criteria*¹. Some of the requirements for C2 level systems that are met by this agency's computing systems include discretionary access controls, sanitation of memory before allocation to another user, and auditing of security related events. Furthermore, many of the agency's computer operating environments implement some security features that are required for the systems that are certified at higher levels.

Because the formal verification and validation process performed by NCSC is such a low level evaluation, it can take over a year to complete. It was requested that the security evaluation methodology to be developed be less expensive and less time consuming than the formal validation and verification. This security evaluation methodology is not intended as a alternative to a formal validation and verification, but as a supplement. Furthermore, the evaluation methodology was to focus on access control measures that are imposed on normal users while operating under routine conditions.

The purpose of this thesis is to present a proven software security evaluation methodology that is generically applicable to any computer operating environment³⁸. The outcome of this evaluation should be a thorough knowledge of the security exposures that exist in the computer operating environment and of the effectiveness of the controls that are in place to prevent these exposures from being exploited. With the knowledge gained from this evaluation, the vulnerabilities of the system should be apparent.

1.3 APPROACH

The proposed methodology is based on a transaction flow analysis approach developed by Charles P. and Shari Pfleeger³⁸. This approach is generally applicable to many certification tasks and has several advantages for software security evaluations³⁸:

Flow orientation. Tracks the security features of a system in terms of the flow of data through the system.³⁸

Organization. Has a useful organization for large and complex systems.²⁵

General applicability. Is applicable to a variety of computing systems, since most systems can be broken down into a sequence of steps and data flows.

Appropriateness for existing systems. Because it operates at the level of functions of systems, it can be applied to an existing system or to an existing system that has been modified. It also can be used to evaluate security of systems that were developed without modern, structured techniques.

This structured and organized approach to security evaluations provides a system security testing methodology that is relatively uniform, expedient, and comprehensive²⁵.

This thesis consists of five sections and the appendices. Section one provides a description of the thesis objectives and the background information on which the work presented here is based. Section two describes the approach used to perform the security evaluation. Detailed descriptions of the parts of the evaluation process and the results of each portion of the evaluation are given in sections three and four. Section five summarizes the overall evaluation effectiveness and lists some recommendations. The appendices include an itemized list of the system controls

and some representations of the mapping work performed as part of the evaluation process.

2. EVALUATION APPROACH

The security evaluation methodology consists of two parts: the Basic Evaluation and the Detailed Evaluation. Figure 1 depicts the evaluation process discussed here. The Basic Evaluation is primarily concerned with the overall functional security posture of the system, which can be determined mostly from the system documentation. The Detailed Evaluation consists of hands-on exercising by authorized users of the individual controls under routine conditions.

The core of the security evaluation methodology discussed is transaction flow analysis. Transaction flow analysis is a systematic, leveled approach to system evaluation in which the system is reduced to a set of conceptually smaller systems, each of which performs one transaction³⁸. As the analysis progresses from general to detailed, the analyst considers smaller portions of the system, resulting in a less complicated and more thorough analysis of the parts of the system. In the Basic Evaluation, the functions of the system are represented as a set of transaction flow diagrams depicting the operations performed on the system objects. The transactions identified in the Basic Evaluation are then used for the hands-on testing performed in the Detailed Evaluation.

2.1 TRANSACTION FLOW ANALYSIS

Transaction flow analysis is referenced in the *Guideline for Computer Security Certification and Accreditation*¹⁰ as a methodology to complement the high-level "completeness" of a Basic Evaluation with detailed, well-understood examples that can focus on a particular aspect of concern. A transaction is typically an event or task of significance to and visible to the user; a transaction flow is defined as a sequence of events involved in the processing of a transaction.

Transaction flow analysis can contribute to the assurance of satisfactory operation of system security. The security functions can be differentiated from

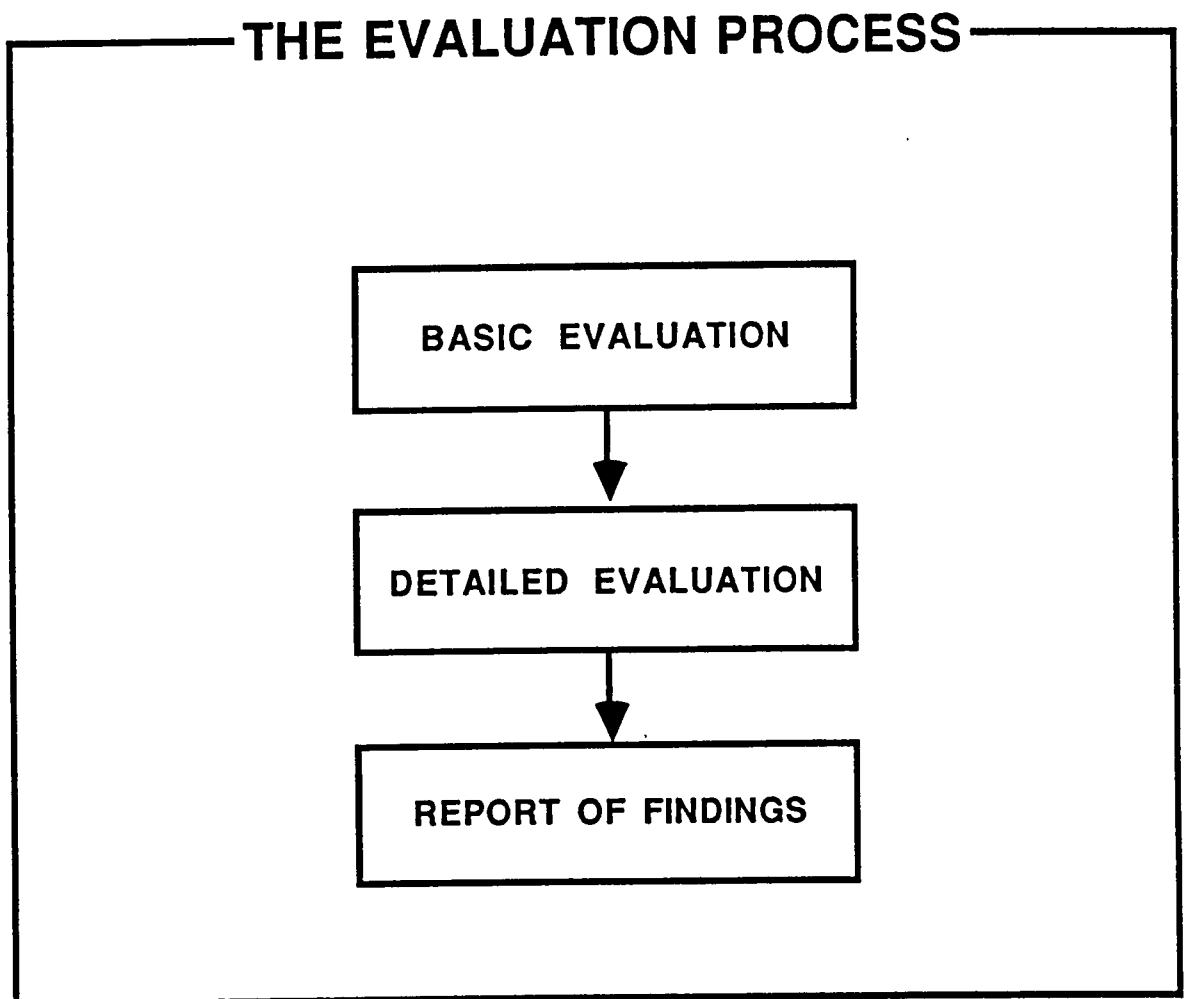


Fig. 1. The Evaluation process.

the operational functions in order to evaluate the effectiveness of the system security. A major advantage of transaction flow analysis is that it is relatively simple to perform, yet it provides comprehensive coverage of the system it describes. Because of the simplicity, the time and resources required to perform system testing with transaction flow analysis are considerably less than those required for the full verification and validation required for system accreditation²⁵. The entire evaluation should take at most three months.

The objective of a transaction flow analysis is to focus on a manageable set of individual operations that can be carefully examined and thoroughly understood. This approach allows the objects, exposures, and controls to be considered in context with respect to each other and to application functions. This evaluation methodology permits a comprehensive consideration of interdependencies and presents a balanced picture of the system security features.

The increased understanding that can result from the utilization of transaction flow analysis as well as its illustrative value make it an important tool in conducting and presenting thorough evaluations. Transaction flow analysis is commonly used in electronic data processing auditing^{2,33} and has been successfully applied to software security certification^{38,40,39,6}.

2.2 SECURITY REQUIREMENTS

A security requirement is an identified computer security need as derived from governmental policy, agency mission, and specific user needs. A security evaluation performed for certification uses the security requirements as the baseline criteria against which the evaluation is performed. The objective of the certification process is to determine whether the computer-system controls satisfy the security requirements. This process is only meaningful if well-defined security requirements exist.

3. BASIC EVALUATION

3.1 BASIC EVALUATION METHODOLOGY

The purpose of the Basic Evaluation is to provide a method for determining the potential system exposures, determining the existing controls associated with each exposure, and assessing the effectiveness of the controls in preventing the exploitation of the associated exposure. Figure 2 illustrates the Basic Evaluation process in a data flow diagram.

3.2 OUTLINE OF STEPS REQUIRED TO PERFORM THE BASIC EVALUATION

The steps of the Basic Evaluation are:

1. **Evaluate the system security requirements.** If the security requirements of the system have not been determined, then a list of requirements must be compiled by an appropriate security official. Once the system security requirements are defined they are evaluated to ensure that the requirements are applicable and that the set is complete.
2. **Identify the exposures.** Given a set of system security requirements, examine the requirements and generate a list of possible exposures.
3. **Identify the controls.** Identify security controls provided by the target system, and build a list of existing controls.
4. **Map exposures and controls.** For each exposure, consider all possible controls and identify the set of applicable controls.
5. **Determine system objects.** Identify all objects in the system. The objects in an operating system are specific to the implementation.

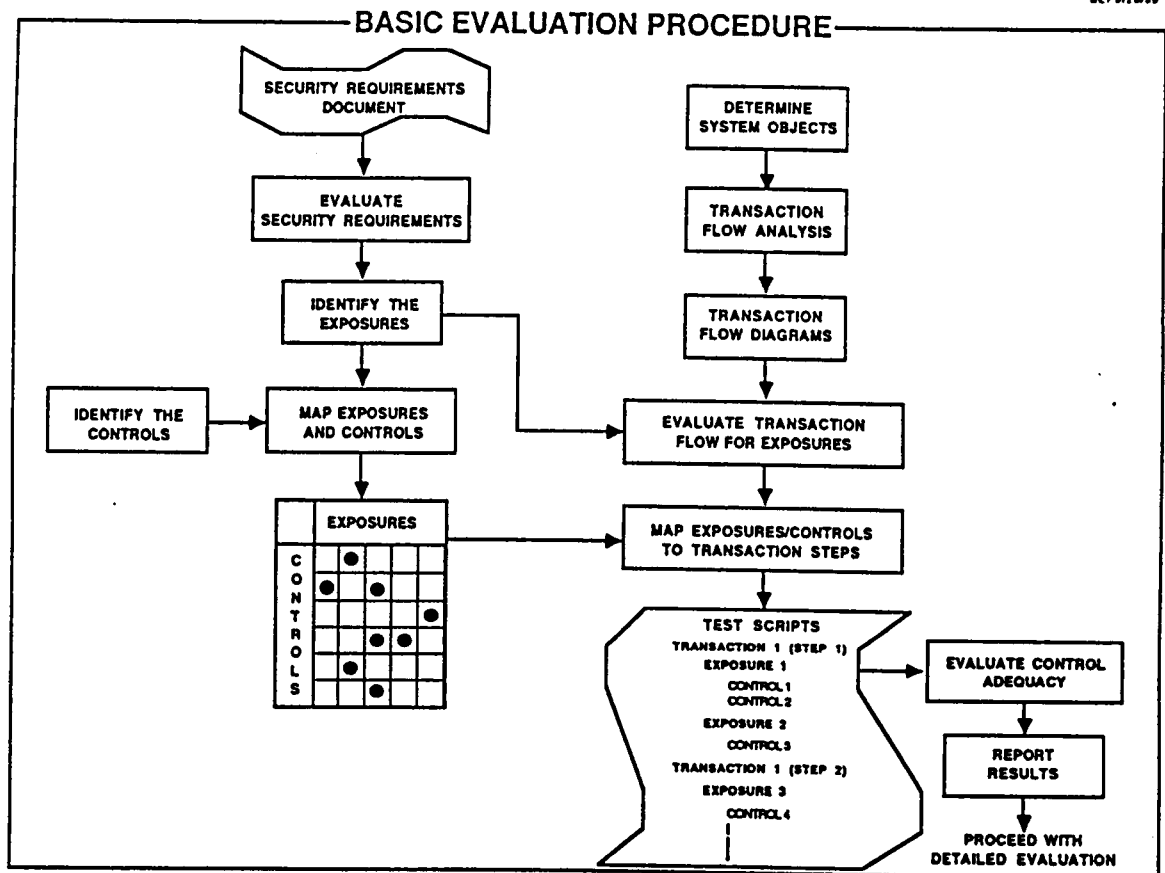


Fig. 2. The Basic Evaluation process.

6. **Determine transaction flows.** Define all possible transactions (operations) that could be performed on system objects. For each data object, depict all transactions on that object with a transaction flow diagram.
7. **Evaluate transaction flow for exposures.** Examine each step of the transaction flow diagram for security exposures, creating a table of exposures associated with each step of the transaction flow. The output of this step is test scripts.
8. **Evaluate control adequacy.** Examine each exposure and the associated set of controls to determine if the set of controls adequately prevents exploitation of the exposure.
9. **Report results.** The results of the Basic Evaluation are compiled to indicate the security weaknesses in the system. The exposures in the system that are determined to be either unprotected or inadequately protected by the existing controls are cited.

The outcome of the Basic Evaluation should be a thorough knowledge of the exposures that exist in the system, the effectiveness of the controls that are provided to protect each exposure, and the unprotected vulnerabilities that exist within the system.

3.3 EVALUATING THE SYSTEM SECURITY REQUIREMENTS

A thorough list of security requirements for the specific site under evaluation must be obtained in order to perform the Basic Evaluation. If the requirements have not been defined, then an appropriate security official must compile a list of applicable system security requirements. The set of requirements is then evaluated to verify that each requirement is applicable to the current operating environment

and that the set of requirements is comprehensive. A comprehensive set of relevant security requirements is essential to perform an effective system security evaluation.

3.4 AGENCY REQUIREMENTS

This section includes a description of the requirements defined for the agency computer operating environments at one site. The requirements are derived from Agency Orders, requirements defined for C2 level systems, and local site requirements.

3.4.1 Security Policy

An explicit well-defined security policy must be enforced by the system. Given identified subjects and objects, there must be a set of rules that are used by the system to determine whether a given subject can be permitted to access a specific object¹.

3.4.1.1 Discretionary Access Controls

Discretionary access controls are responsible for enforcing a need- to-know policy. This type of control requires that access to certain objects be restricted to those users with a formal need to know, regardless of user clearance, and thus the following requirements are enforced:¹.

1. The system shall provide discretionary access control mechanisms which shall define and control access between subjects and objects, thereby protecting objects from unauthorized access.
2. Authorized users shall have the capability to define access rights to objects that they own.
3. The discretionary access control mechanism shall provide operational security controls to limit propagation of access rights.

4. The discretionary access control mechanism shall enforce a formal need to know policy in which the discretionary access control mechanism shall, by default, provide that objects are protected from unauthorized access.
5. File access permissions shall default to *no access* except for the owner of a file for classified and sensitive computers. On nonsensitive computers, file access permissions shall default to READ and EXECUTE access for all users.
6. Directory access permissions shall default to *no access* except for the directory owner for classified and sensitive computers. On nonsensitive computers, directory access permissions shall default to READ and EXECUTE access for all users.

3.4.1.2 Object Reuse

The operating system shall ensure that residue from user programs in both batch and on-line modes are cleared or written over before memory and other storage are allocated by the system for use by other user programs¹.

3.4.1.3 Labels

Access control labels must be associated with objects in the system. In order to control access to information stored in a computer system it must be possible to mark every object with a label that reliably identifies the object's sensitivity level, and/or the modes of access accorded those subjects who may potentially access the object¹. The requirements for labeling include the following:

1. The system must maintain reliable (assigned or changed only by an authorized security administrator) labels on all objects to reflect the degree of protection required for the object and to reflect the authorization of a subject (user or processes) to access the object.

2. The system shall be capable of associating security labels with user terminals and other user-employable devices before allowing use of system resources.
3. A change in the classification level or sensitivity label of an object shall be controlled so that such a change can be made only by an authorized system security administrator.
4. Labels shall be associated with electronic information being exported.
5. Human readable (text) labels shall be associated with human readable output.
6. Security related software shall request and receive the security level of data being input to the system.

3.4.1.4 Mandatory Access Controls

There must exist a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity¹.

Requirements in this area include:

1. The system shall enforce mandatory access control policies over all subjects and objects under its control.
2. The mandatory access controls shall include access controls based on the classification level of the data and the security clearance level of the user, i.e., no individual shall have access to classified data for which that person does not have the appropriate clearance.
3. Subjects and objects shall be assigned sensitivity labels that are a combination of hierarchical classification levels and non-hierarchical categories, and the labels shall be used as a basis for mandatory access control decisions.

4. The mandatory access controls shall be supported by a mechanism that ensures the integrity of all data in the system, including labels that represent the sensitivity levels.
5. All users shall have a predefined user profile which shall determine their need to access objects. This user profile shall determine access rights to all objects in the system.
6. The system shall ensure that user profiles can be established and changed only by the system security administrator.

3.4.2 Accountability

In order to mediate access control, there must exist a means to identify each individual subject uniquely and the objects that the subject is to be allowed to access¹.

3.4.2.1 Identification and Authentication

1. The system shall require users to identify themselves before revealing any system information or allowing any access to system resources.
2. The system shall be able to enforce individual accountability by providing the capability to identify each individual user uniquely.
3. All users shall require individual user IDs and nondisplayable passwords.
4. Passwords shall not be displayed at login.
5. The system shall protect authentication data including passwords so that it cannot be accessed or changed by any unauthorized subject.
6. Authentication data that is used to verify the identity of individual users (e.g. passwords) shall not be stored in plain text within the computer system.

7. The authentication of users shall employ a one-way encryption technique for passwords.
8. Initial passwords shall be assigned and distributed to users of the system.
9. If users are allowed to change their passwords, they are required to use the password generation utility to generate the new password.
10. Processes shall be identifiable to the host processor for control of I/O and for disconnection when not in use.
11. The time and date of the last successful login will be displayed at login.

3.4.2.2 Audit

Selective audit information pertaining to the security aspects of the system must be reliably maintained in order to trace security violations to the responsible party¹.

The requirements in this area are as follows:

1. The system shall ensure the capability of associating the individual user identifier with all auditable actions taken by that individual.
2. For each recorded event, the audit record shall identify:
 - a. Date and time of event
 - b. Success or failure of event
 - c. Type of event
 - d. User or process identification
 - e. For identification/authentication events the origin of the request (e.g., terminal ID) shall be included in the audit record.
3. For events that introduce an object into a user's address space and for object deletion events, the audit record shall include the name of the object and the object's security level.

4. At a minimum, the following events shall be audited:
 - a. All attempts to access system resources, whether or not the attempt is successful
 - b. System faults and restarts
 - c. Actions of system security administrators and computer operators
 - d. User inactivity timeouts (thresholds shall be adjustable by the system security administrator)
 - e. Modification to user's authentication profile
 - f. Use of identification and authentication mechanisms (all logins)
 - g. Introduction of objects to subjects (file open, file close)
 - h. User logoff activity
 - i. Creation and deletion of objects
5. A selective auditing capability shall be provided to enable the system security administrator to select the events to be audited.
6. The system shall provide protection to the events logs and audit files to prevent changes or deletions.
7. The system shall provide the system security administrator summary reports of user profiles and changes to the user profiles.

3.4.3 Assurance

In order to assure that the defined requirements are enforced by a computer system, there must be some identified and unified collection of hardware and software controls that perform those functions. These mechanisms are typically embedded in the operating system and are designed to carry out the assigned tasks in a secure manner¹.

3.4.3.1 Operational Assurance

The trusted mechanisms that enforce the basic requirements of the system must be continuously protected against tampering and/or other unauthorized changes¹.

Requirements include:

1. The operating system shall be maintained in a domain for its own execution that protects it from external interference or tampering (e.g. by modification of its code or data structures).
2. The operating system shall enforce the READ, WRITE, EXECUTE, and DELETE privileges of a user with respect to any given file.
3. Message stream modification countermeasures shall be used to deter and detect changes to sensitivity labels associated with data transported through the system.
4. The operating system shall control the continued separation of user and privileged modes so that a program operating in user mode is prevented from unauthorized use of privileged functions.
5. System hardware and software features shall exercise control over the memory locations to which a user program has access.
6. The operating system shall restrict memory dumps so that individuals are exposed only to the information or programs in memory for which they have access authority.
7. One class of instructions reserved for exclusive use of the operating system and one class usable by both the operating system and the user's programs shall be provided.
8. A set of privileged instructions shall be provided which enable the operating system to restrict access of users to input/output devices, memory

management functions, and all information which defines the execution characteristics of user processes.

9. The hardware and software shall be designed so that all machine instructions and operation codes, both privileged and user, produce results in a predefined set of responses.
10. Error detection shall be performed on each FETCH cycle in order to avoid erroneous commands, data and/or addressing which may result in unauthorized or incorrect action by the computer.
11. The operating system alone shall control input/output operations, memory management functions, and maintenance of execution state information.
12. Registers shall be protected by error detection or redundancy checks to provide secure operation of the hardware.
13. Initial login and authentication shall be affected via a direct communication path between the user login input device and the security related software of the system.
14. Login attempts by any user shall be limited. The number of attempts allowed is a system parameter under the control of the system administrator.
15. If the number of login attempts exceeds the number allowed, the user shall be locked out by the system. System reinitialization shall not reinitialize the lockout state of any user.
16. User authorization after lockout shall be reinitiated only by a system administrator.

17. Interactive processes shall be automatically terminated after "n" minutes of inactivity. The timeout interval shall be flexible as defined by the system security administrator.
18. The capability to lockout specific terminals on the system on a selective basis by the system administrator shall be provided.
19. The system shall be able to deny or restrict users to certain system resources (workstations, processors, tape drives, etc.).
20. A power-up of the system shall result in reinitializing all terminals except those locked out by the system administrator. Also, the power-up of a terminal shall reinitialize the terminal unless it is locked out by the system administrator.
21. Hardware and/or software features shall be provided that can be used to validate periodically the correct operation of the on-site hardware and firmware elements of the system.

3.4.3.2 Life-Cycle Assurance

The actual development of the operating system is controlled by the product vendor. Adherence to any life-cycle requirements before installation in an agency operating environment can be confirmed only by the agency security personnel. Requirements include:

1. Security Testing - The security mechanisms of the system shall be tested, and found to work as claimed in the system documentation prior to production use.
2. Configuration Management - Any local modifications made to the operating system code shall be controlled by a configuration management system.

3.4.4 Documentation

The basis for trusting the system control mechanisms in their operational setting must be so clearly documented that independent examination of the evidence can be made to evaluate their sufficiency¹.

3.4.4.1 Security Features User's Guide

A single summary, chapter or manual in user documentation shall be provided that describes the protection mechanisms provided by the system, guidelines on their use, and how they interact with one another shall be provided¹.

3.4.4.2 System Security Administrator's Manual

A system security manual, addressed to the system security administrator, shall be maintained¹. The requirements for this manual are as follows:

1. The system security manual shall present cautions about functions and privileges that should be controlled when running a secure system.
2. The system security manual shall also describe the procedure for configuring the automated security functions of the system and other administrative procedures for system security.
3. The procedures for examining and maintaining the audit files (including the detailed audit record structure for each type of audit event) shall be given.
4. The manual shall describe the operator and administrator functions related to security, including changing of the security characteristics of a user.
5. The manual shall provide guidelines on the consistent and effective use of the protection features of the system; how they interact; how to generate securely a new system and procedures; warnings and privileges that need to be controlled in order to operate the system in a secure manner.
6. Configuration policies and procedures concerning system administration shall be described.

3.4.4.3 Test Documentation

A document shall be provided that describes the system test plan, the test procedures that show how the security mechanisms were tested, and the results of the security mechanisms' functional testing¹.

3.4.4.4 Design Documentation

A document shall be provided that describes the designer's philosophy of protection and an explanation of how this philosophy is translated into the system. The interfaces between the system modules shall be described¹.

3.4.5 Administrative Requirements

Additional administrative requirements are necessary to encompass a comprehensive security policy. These include:

1. Procurement shall be limited to those companies who are reputable, bonded, and have signed certification statements declaring their products to be free of subversive code (trojan horses, time bombs, viruses, worm-hole routines, etc.). Where local modifications are required, source code shall be furnished or the appropriate hooks provided.
2. An Automatic Data Processing (ADP) Security Plan shall be submitted to the proper accrediting official, the Computer Security Operations Manager (CSOM), for approval before classified information can be processed on a computer system. Further, if the system is a network of components not under the jurisdiction of the same CSOM the accrediting official is the Computer Security Program Manager (CSPM). This effort must be coordinated with the site computer security group.
3. A Computer Protection Plan shall be submitted by the Computer Protection Program Manager (CPPM) to the appropriate site management for approval

before unclassified sensitive information can be processed on a computer system.

4. Periodic risk assessment reviews shall be conducted for unclassified sensitive computers and classified computers in accordance with agency orders.
5. Backup procedures shall be implemented as appropriate to ensure minimal loss of data in the event that data restoration from backup media is necessary. One copy of the data backups shall be kept in a different location.
6. A contingency recovery plan shall be developed and periodically reviewed to ensure that the plan is kept current.
7. All vendor supplied or publicized accounts shall be disabled or the passwords changed.
8. The number of users having access to privileged accounts shall be held to a minimum. Up-to-date lists of those persons having access to privileged accounts and the accounts themselves shall be submitted to the appropriate security authority. This list shall include users with the ability to grant privileges.
9. There shall be no non-encrypted dial-up access or non-encrypted dial-out capability for classified computers.
10. There shall be no proxy access to privileged accounts on the system.
11. For all dial-up connections, the session shall automatically be terminated in the event a terminal is disconnected. This prevents another user from connecting to the disconnected session.

12. All dial-in/dial-out capability shall be under the control of a central management of dial-up access. [Note: This is an installation specific requirement in some agency installations.]

3.5 IDENTIFYING THE EXPOSURES

The system exposures are identified from an analysis of the system security requirements. The security requirements provide a base from which the system vulnerabilities can be determined and a list of exposures can be compiled. For each requirement, the possible exposures associated with that requirement are identified. A generic requirements/exposures list generated for the VAX/VMS environment under evaluation is included in Appendix A.

3.6 IDENTIFYING THE CONTROLS

Controls are specific to the system being evaluated. Most of the controls provided by a specific system can be determined from the system documentation. However, many of the security features provided by a system are optional and must be enabled by the system manager to qualify as operational controls.

The controls list should further include any controls beyond those provided by the operating system that have been implemented. Additional controls would include administrative controls that are instituted to comply with agency orders and software controls that have been developed locally to satisfy site-specific requirements. The installation-specific controls can be identified by the local system administrator. A itemized description of the controls implemented in a VAX/VMS environment at one agency installation is included in Appendix B.

3.7 COMPUTER OPERATING ENVIRONMENT CONTROLS

Many of this government agency's computer operating environments use the VAX/VMS operating system. At the time this security evaluation was conducted the most current VAX/VMS version was 5.0. Therefore, this version was chosen for the evaluation process. The operating environment controls consist of those

provided by the VMS operating system, the locally implemented controls, and the agency administrative controls. The following sections describe these controls.

3.7.1 VMS System Access Controls

VAX/VMS 5.0 provides seven classes of logins: Local, Dialup, Remote, Network, Batch, Detached, and Subprocess. These logins are differentiated into two types: interactive or noninteractive. An interactive login consists of a series of steps in which the system prompts for information that must be provided by the user. Noninteractive logins are performed by the system with no input required from the user²². The classes and types of logins provided by the VMS operating system are shown in Table 1.

Table 1. VMS logins.

CLASS	TYPE
local	interactive
dialup	interactive
remote	interactive
network	noninteractive
batch	noninteractive
detached	dependent on parent process
subprocess	noninteractive

Logins can be restricted by class to include or exclude any of the classes listed above. Further restrictions can be imposed on a class of login to restrict logins of that class to certain times of the day and certain days of the week²².

During an interactive login, several messages can be displayed at various points in the login process. The announcement message is the first visual indication that the login process has begun. This message usually identifies the node name to which

the login is being attempted and immediately precedes the *Username* prompt. The announcement message can be defined or disabled by the system administrator by redefining the logical name SYS\$ANNOUNCE in the system startup procedure²².

If the login process completes successfully, there could be messages displayed to indicate that there is a disconnected job available for reconnection. This message is displayed when a previous login was interrupted and the terminal was set up as a virtual terminal. The user is given the option of reconnecting to the disconnected job and returning to the state just before the interruption occurred. If the user does not reconnect to the disconnected job or delays too long, then the process times out and is terminated²².

The next message that can be displayed is the welcome message, indicating the version of VMS that is running and possibly the node name. The system administrator can either alter the welcome message or disable it entirely²².

An optional set of messages providing information about the last successful login can be displayed just after the welcome message. These last login messages are either enabled or disabled as a group. If enabled, the following messages can appear²²:

1. Last successful interactive login - displays the time of the last completed local, dialup, or remote login.
2. Last successful noninteractive login - displays the time of the last completed batch or network login.
3. Number of login failures - displays the number of unsuccessful login attempts since the last successful login.

VMS provides four types of accounts. Open accounts do not require passwords, hence the user is not prompted for a password. Captive accounts limit user

operations and may or may not require a password. A third type of account always requires a password and prevents the user from changing the password. The fourth type of account requires a password that is changed periodically by either the system administrator or the user. When an account is created the system administrator can specify an expiration time for the account. After the expiration time has passed the user receives an authorization failure message when a login is attempted²².

For accounts requiring passwords to be entered by the user, the terminal does not echo passwords as they are entered. A one-way encryption algorithm is applied to all passwords before they are stored in the system. When a user enters a password, the same algorithm is used to process the entered password before matching it with the stored value²².

There are three types of passwords that may be required: system passwords, primary user passwords, and secondary user passwords. The use of system passwords is optional and can be used to control access to particular terminals. User passwords are used to control access to or from particular accounts. The secondary user password can be required in addition to the primary user password in order to access an account. There is a limited amount of time allowed for the entire login process. If the required passwords are not successfully entered within this time period, the login process will time out²².

Several optional restrictions can be imposed on password use. The system administrator can either prohibit users from changing their own passwords or require the use of the password generator utility when a user changes a password. VMS includes a password expiration feature that forces users to change their passwords by a specified expiration date or the account will be disabled. Also, a minimum password length can be imposed²².

VMS provides a secure terminal feature to protect against password grabber programs. This is an optional feature that ensures that the VMS login program is the only program able to receive the user name and password entered in the login process²².

VMS includes control measures to deter unlimited trial-and-error attempts to login to the system. An optional break-in evasion feature is available that allows the system administrator to specify a maximum number of unsuccessful login attempts that can be attempted within a time interval. If the maximum number of unsuccessful login attempts is exceeded in the time interval, then break-in evasion is activated. Once break-in evasion is activated, even the correct user name and password results in an authorization failure for a time interval. The monitoring can be based on user name or a combination of user name and terminal. If both user name and terminal are specified, then evasive action is triggered for only the terminal where the maximum number of login attempts was exceeded. The break-in evasion feature can be configured to disable a user for which a break-in has been detected. VMS disables user accounts by executing the AUTHORIZE utility with the /FLAGS=DISUSER qualifier²².

Control measures are provided by the VMS operating system for the use of dial-up terminals. The system administrator can specify a maximum number of times that a user can attempt to enter a correct password before the dial-up connection is automatically terminated. If the number of attempts is exceeded, then the connection is broken, and the user must redial for another connection. The /HANGUP option of the LOGOUT command can be used to break the connection automatically after a logout has completed. This requires another user to redial the number for a connection²².

Proxy logins are a type of network login that allow users to access the system across a network without specifying a user name or password in an access control string. This feature reduces the risk of exposing the authentication data since the user name and password are not entered by the user, and this data is not transmitted across the network²².

3.7.2 VMS Object Access Controls

The VMS operating system offers two primary protection mechanisms²². One type of protection, called UIC-based protection, is based on the user identification code (UIC) and controls access to objects according to the user categories SYSTEM, OWNER, GROUP, and WORLD. The other primary protection mechanism employs a more refined level of protection using access control lists (ACL). ACLs can grant or deny access to objects independent of the UIC. Also, users can hold certain privileges that affect user access to objects. The interaction between ACLs, the UIC-based protection, and the user privileges determine whether access to an object is granted or denied. The VMS operating system performs the following steps to determine whether a user is allowed to access an object²²:

1. If an object has an associated ACL, the system uses that ACL to determine whether the user should gain access to the object. If the ACL grants the requested access to the user, access is given and all further testing stops. If the ACL denies access, the system uses the SYSTEM and OWNER fields of the UIC-based protection to determine if the user is allowed access. If the ACL does not explicitly grant or deny the user access, the system uses UIC-based protection to determine access.
2. If an object does not have an associated ACL, the system uses the UIC-based protection to determine access.

3. The GRPPRV, SYSPRV, READALL, and BYPASS privileges amplify the privilege holder's access to objects under certain circumstances.

VMS maintains a rights database that contains a list of identifiers and the users specified as holders of those identifiers. By assigning users as holders of identifiers, groups of users can be defined independent of their UIC, allowing more refined access control. Each time a user logs in to the system the system creates a process rights list containing all of the rights associated with the user process. This process rights list is used to determine the objects that a user is allowed to access. The initial rights database is created at system installation and contains the names of the reserved system identifiers and one identifier for each authorized user. The user identifiers are called UIC identifiers and are associated with the user's UIC and user name. The rights database also contains an identifier for each user group name. As users and groups are added to the system, identifiers are automatically added to the rights database. However, additional identifiers can be created to group users independent of their UIC. These identifiers are maintained manually by the system administrator²².

3.7.2.1 UIC-based Protection

All users and objects in the system have an associated UIC. The UIC of an object is defined to be the UIC of the object owner. The objects protected by UIC-based protection have a protection mask that defines which types of access (READ, WRITE, EXECUTE, DELETE) are allowed to each user class (SYSTEM, OWNER, GROUP, WORLD). The relationship between the UIC of the user and the UIC of the object controls the access to the object. The system checks for access granted to user categories in the following sequence: OWNER, WORLD, GROUP, SYSTEM. Each user account is established with a default protection code for all

files the created in the initial top-level directory. Also, the user is assigned to an initial group when an account is created²².

The UIC is stored as a 32-bit value that represents a group number and a member number. The 32-bit numeric UIC is kept in the system rights database, which is a file containing information pertaining to the access rights of the subject. The method of storing alphanumeric UICs dictates that member names must be unique, and that no member can participate in more than one group. UIC-based protection is imposed on objects such as files, directories, volumes, and record oriented devices such as terminals, printers, and mailboxes²².

3.7.2.2 ACL Protection

ACLs provide a more precise means of defining access to objects than the UIC-based protection. Users can be grouped independent of their UIC. ACLs can be defined for the following objects²²:

1. Devices
2. Files
3. Group global sections
4. System global sections
5. Logical name tables
6. Queues

An ACL consists of access control entries (ACE) that either grant or deny access to an object. Each ACE specifies a user or group or users and the type of access permitted. The following are the three types of ACEs involved with security²²:

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

2. Default Protection ACE - Defines the default protection for a directory so that the protection can be propagated to the files and subdirectories created in that directory.
3. Security Alarm ACE - Provides an alarm message when an object is accessed to alert managers to possible security threats.

An identifier ACE can be used to grant or deny access to groups of users who are holders of the identifier. Use of this feature allows users to be grouped in a more flexible manner than the UIC-based group definitions. Identifiers specified in an ACE can be of the following three types²²:

1. UIC identifiers - depend on the user identification codes (UICs) that uniquely identify each user on the system.
2. General identifiers - defined by the security manager in the system rights database to identify groups of users on the system.
3. System-defined identifiers - describe certain types of users based on their use of the system. For example, BATCH, NETWORK, DIALUP, INTERACTIVE, LOCAL, and REMOTE would correspond directly to the type of login the user executed.

ACLs can be created by the system by default, by the system security administrator for specific objects, and by users to protect their own files²².

3.7.2.3 Attributes and Privileges

Identifiers are associated with attributes that are specified when the identifier is added to the rights database or when the identifier is granted to a user. These attributes are as follows²²:

1. Resource - allows holders of the identifier to charge disk space to the identifier.

2. Dynamic - allows holders of the identifier to remove the identifier from the process rights list.

CONTROL access is a special kind of access that can be specified in an ACL and is automatically granted to users of certain user categories when UIC-based protection is evaluated. For example, the creator of a file is automatically given CONTROL access when the file is created. This ensures that the creator of a file retains control of the file even if the ownership is changed. This type of access grants the accessor all of the privileges of the object's actual owner²².

VMS features four system privileges that affect user access, regardless of the access dictated by UIC-based or ACL protection. These privileges are²²:

1. SYSPRV - A user with SYSPRV receives the access accorded to users in the SYSTEM category.
2. GRPPRV - A user with GRPPRV privilege whose UIC group matches the group of the owner of the object receives the same access accorded to users in the SYSTEM category.
3. BYPASS - A user with BYPASS privilege receives all types of access to the object, regardless of its protection.
4. READALL - A user with READALL privilege receives READ and CONTROL access to the object, regardless of the UIC-based and ACL protection. The user also receives any additional access granted in other fields of the protection code.

The execution of some images require certain privileges. A user cannot execute an image that requires a privilege the user does not have unless the image is installed as a known image with respect to the privilege required²².

3.7.2.4 Restricting User Capabilities

The system manager can set up specific terminals to be accessible to the SYSTEM account only. This is accomplished via the SYSGEN parameters TTY_DEFPROT and TTY_OWNER. These parameters define the default protection and owner for terminal devices. Terminals can be made accessible to users as application terminals by adjusting the device protection code and the owner UIC of the terminal or through the use ACLs²².

The use of VMS DCL commands can be controlled by removing or modifying the DCL command definitions and rebuilding the DCL tables. Use of the modified table can be specified for a user with the /CLITABLES qualifier in the user authorization file. A user can be restricted to only the modified DCL table with the /FLAGS=DEFCLI qualifier. Also, ACLs can be imposed on the system program files in the directories SYS\$SYSROOT:[SYSEXE] and SYS\$SYSROOT:[SYSLIB] to restrict use of DCL commands²².

3.7.3 File and Directory Ownership

Ownership privileges allow a user to set or change the ownership of an object. A user must satisfy one of the following conditions to have ownership privileges to a file or directory²²:

1. Must hold the resource attribute to the identifier that owns the file.
2. Must qualify as a member of the SYSTEM user category, hold SYSPRV or BYPASS privilege, or hold a UIC that matches that of the owner of the volume containing the file or directory.
3. Must hold the GRPPRV privilege while also holding a UIC in the same group as the object owner.

If ownership is not specified when the file or directory is created, then VMS assigns a default owner. To establish default-file ownership, an attempt is first made

to propagate the ownership from previous versions of the file. This propagation of ownership can be successful only if the user creating the file has ownership privileges to the previous version. If the propagation of ownership from a previous version of the file fails, then VMS attempts to propagate ownership from the parent directory. Again, the user creating the file must have ownership privilege to the parent directory. If both attempts to propagate the ownership fail, then the owner of the file is defined to be the same as the creator of the file²².

If no owner is specified when a directory is created, then VMS assigns a default owner. If the directory name is in alphanumeric or subdirectory format and the user creating the directory has ownership privileges to the parent directory, then ownership defaults to the owner UIC of the of the parent directory. If the directory name is in alphanumeric or subdirectory format and the user does not have ownership privileges to the parent directory, then ownership of the subdirectory defaults to the UIC of the process issuing the command. If the directory name is in UIC format, the ownership defaults to the UIC in the directory name²².

3.7.4 Volume Ownership

Ownership of a volume is established when the volume is created. Only users who satisfy one of the following criteria can change ownership of a volume²²:

1. Qualify as a member of the SYSTEM user category.
2. Hold the SYSPRV privilege.
3. Own the volume.

3.7.5 VMS Protection Defaults

The protection code of a directory defaults to that of the parent directory. The protection code of a top-level directory defaults to that of the volume master file directory. A default protection ACL can be used to establish a different default protection scheme. Since subdirectories inherit the ACLs of the parent directory,

the default protection code established in the ACL will be propagated throughout the tree unless the NOPROPAGATE option is specified²².

If a protection code is not specified when a file is created, then the system will attempt to propagate the protection code from a previous version of the file. If there is no previous version of the file and the directory where the file is to be created has an ACL that includes a default protection ACE, then the protection specified in the ACL is used. If the directory has no default protection ACE, then the default process protection is used. The default process protection is determined from the SYSGEN parameter RMS.FILEPROT during the login process, but this value can be overridden after the login is completed²².

When a new version of an existing file is created, then the new file inherits the ACL of the previous version without the ACEs that are marked with the NOPROPAGATE option. If a new file does not have a previous version, then all ACEs in the directory ACL that are marked with the DEFAULT option are copied to the new file with the DEFAULT option removed²².

When a user creates a file specifying another user as the owner, an ACE is automatically added to the file ACL granting CONTROL access, as well as all types of access accorded to the owner, to the creator. This features guarantees that the creator of a file retains control of the file, regardless of other defaults²².

3.7.6 Reuse

VMS provides control measures to prevent disk scavenging. An erasure pattern of a repeated sequence of bits can be applied to write over the disk area occupied by deleted files. The system manager can specify use of the erasure pattern on a per volume basis. Also, users can request application of the erasure pattern on a file-by-file basis²².

Highwater marking is a technique used to prevent users from reading file space beyond the areas where they have been permitted to write. The outer limit of written space on the file is referred to as the high water mark for that file. This technique prevents users from scavenging the disk for data written by other processes. VMS implements highwater marking on sequential, exclusively-accessed files, such as files output from text editors, compilers, and linkers²².

A technique known as erase-on-allocate is used for indexed and shared sequential files instead of true highwater marking. Erase-on-allocate means that blocks of disk space are erased as they are allocated to a process. Although highwater marking is enabled by default when a volume is initialized, the system administrator can disable this feature²².

Reuse of system memory pages is protected by the VMS memory management subsystem. The memory management subsystem ensures that residue from previous processes is expunged prior to allocation of memory to another process⁴.

3.7.7 Operational Security

The VAX hardware architecture provides for processor access modes: user, supervisor, executive, and kernel. These access modes are used to determine which instructions a process is allowed to execute and which memory locations those instructions can access. Thus, the access modes protect the VMS system from processes executing in user mode, as well as from separate parts of the operating system according to their degree of privilege²⁷.

The memory management system prevents modification of the code and data used by more privileged access modes by less privileged modes. Further, the VMS memory management architecture protects a process' address space from code that is running in the context of another process. Although system virtual address space

is addressable by all processes, page-by-page protection controls access to specific virtual pages for certain access modes²⁷.

The VMS operating system controls allocation of its system-wide resources, such as nonpaged dynamic memory and page file space, through the use of quotas and limits. These process attributes are assigned when the process is created. By restricting these system resources, VMS exercises control over the resource drain that can be exerted by a single process²².

A utility called the User Environment Test Package (UETP) is provided to test the performance of system hardware and software. This utility runs test exercises to simulate user programs that access disk files while checking file system inconsistencies. UETP also performs some minor diagnostics. This utility does not attempt to diagnose the cause of errors; it simply reports detected errors⁴.

The VMS ERRFMT process records the system errors in the file SYS\$ERRORLOG:ERRLOG.SYS. This file contains hardware device errors, error correction code errors, interrupt exception conditions, machine checks, and software bugchecks. Error correction codes repair single-bit errors and detect double-bit errors⁴.

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 the correct function of the microcode and the data path circuitry used in the address translation logic of the virtual memory subsystem⁴.

3.7.8 Auditing

VMS provides tools to detect security violations and system abuse. A security operator terminal can be defined for monitoring security-related events. The system security administrator can select the types of events to be audited and direct the

system to send an alarm message to the security operator terminal when an audited event is detected. The following events can be audited²²:

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

The alarms sent to the security operators' terminal reveal the name of the perpetrator, the method of access, time of access, and the privilege used to to gain access to an object. VMS writes information resulting from security alarms into the security operator's log. The SECAUDIT.COM command procedure can be used to extract selective information from this file. The security administrator requires the SECURITY privilege to enable security auditing or to set up a security operator terminal²².

Actions taken by trusted users (e.g., system administrators, security administrators, etc) can be audited by use of terminal session auditing. Terminal session auditing can be enforced with captive accounts and the appropriate command procedures. The users for whom session auditing is enforced must first

log into a captive account, then into their own account. The entire terminal session is saved in a log file²².

The VMS 5.0 system accounting log is another useful tool for tracking system activity. This log contains records of all system job terminations including all interactive, batch, and network jobs, as well as print jobs and other process terminations. All or selected images can be monitored with the accounting facility²².

3.7.9 Documentation and Life Cycle Controls

VMS version 4.3 has been certified as a C2 system⁴. Therefore, it is assumed that the documentation and life cycle requirements for a DOD C2 level system have been met.

3.7.10 Local Site Controls

An automatic timeout mechanism has been locally implemented to terminate idle processes. If a process has been inactive for a specified period of time, then the process is terminated.

Initial user passwords are assigned and distributed by an authorized system security official. Users are not allowed to change their own passwords. Replacement passwords are assigned and distributed to users periodically.

All vendor supplied or publicized accounts are disabled or the passwords are changed. Also, the number of users having access to privileged accounts is held to a minimum. Up-to-date lists of those persons having access to privileged accounts and the accounts themselves are submitted to the appropriate security authority. This list includes users with the ability to grant privileges.

3.7.11 Agency Administrative Controls

Non-encrypted dial-up access and non-encrypted dial-out capability is not allowed on computers used for processing of classified or sensitive information. For all dial-up connections, the session shall automatically be terminated in the event a

terminal is disconnected. Further, all dial-in/dial-out capability shall be under the control of a central management of dial-up access.

Procurement is limited to those companies who are reputable, bonded, and have signed certification statements declaring their products to be free of subversive code (trojan horses, time bombs, viruses, worm-hole routines, etc.) Where local modifications are required, source code or the appropriate hooks provided must be provided.

An ADP Security Plan is submitted to the proper accrediting official and the Computer Security Operations Manager (CSOM) for approval before classified information is processed on a computer system. Further, if the system includes a network environment that crosses operations office geographic boundaries, then the plan is approved by the Computer Security Program Manager (CSPM). This effort is coordinated with the site computer security group.

A Computer Protection Plan is submitted by the Computer Protection Program Manager (CPPM) to the appropriate site management for approval before unclassified sensitive information is processed on a computer system.

Periodic risk assessment reviews are conducted in accordance with Agency Orders for unclassified sensitive computers and classified computers. Backup procedures are implemented as appropriate to ensure minimal loss of data in the event that data restoration from backup media is necessary. One copy of the data backups is kept in a different location. Furthermore, a contingency recovery plan has been developed and is periodically reviewed to ensure that the plan is kept current.

3.8 MAPPING EXPOSURES AND CONTROLS

For each exposure identified in the system, the set of applicable controls provided to prevent exploitation of the exposure is identified. The relationship between

exposures and controls can be one-to-one, one-to-many, or many-to-many. Ideally, there should be at least one control provided for each exposure in the system. The mapping of exposures and the applicable controls can be represented as shown in Appendix C.

3.9 DETERMINING SYSTEM OBJECTS

Objects in an operating system consist of the system resources and data structures for which control of shared access among subjects is the responsibility of the operating system. The objects in an operating system are specific to the implementation. Different operating systems are implemented with different data structures. However, most of the data structures used by an operating system are protected by the same access control mechanisms. For example, objects defined in the VAX/VMS operating system for which access control mechanisms have been implemented include the following:

1. Disk files and directories
2. Queues
3. Logical name tables
4. Global sections for shared use
5. Mailboxes
6. Event flag clusters
7. Devices

Access to these VAX/VMS objects is determined via access control lists (ACLs), user identification codes (UICs), and user privileges.

3.10 DETERMINING TRANSACTION FLOWS

A transaction in a computer operating environment is a sequence of steps that must be performed to access an object. A transaction flow diagram is a graphical

representation of a transaction in the form of a data flow diagram. For each combination of object and transaction, a transaction flow diagram is drawn to depict the possible operations performed on an object.

The following types of access are defined for VAX/VMS objects: READ, WRITE, DELETE, and EXECUTE. These access modes can be viewed as transactions that can be performed on the operating system objects. VMS applies different meaning to the access types depending on the object. For example, WRITE access to a file allows a user to write to or modify the file, while WRITE access to a queue allows a user to submit jobs to the queue. Although the context of the transaction may vary with the object, the access controls imposed on a transaction are based on access control mechanisms that are the same for all the objects. Thus, the transactions for these four access types can be generalized to include all of the system objects.

Figures 3 through 6 are the four top-level transaction flow diagrams for the VMS operating system. The diagrams show a generalized schematic of each of the four transactions READ, WRITE, EXECUTE, and DELETE. These transaction flow diagrams are all very similar until the "access granted" step, where access is granted to perform one of the four basic transactions. The transaction flow diagrams are very similar because the functions performed on objects have been generalized to collectively include the system resources and data structures used by the operating system that require access control protection.

Figures 7 and 8 detail the "Login" and "Access Control" subsystems of the basic transaction. These figures are adaptations of data flow diagrams presented in the VMS System Documentation ^{22,22,27,23,21}. These subsystems are detailed to facilitate presentation of exposures and controls in the VMS operating system.

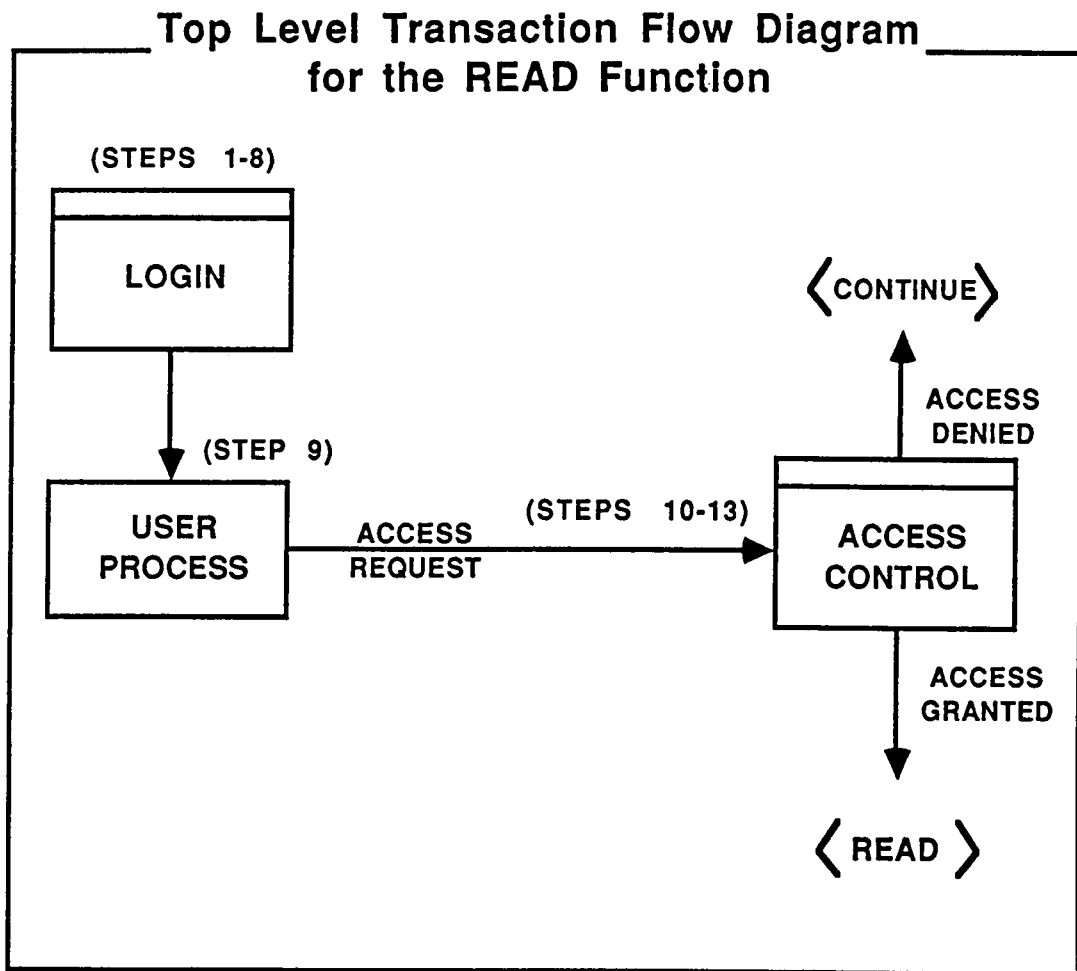


Fig. 3. Transaction Flow Diagram for the READ function.

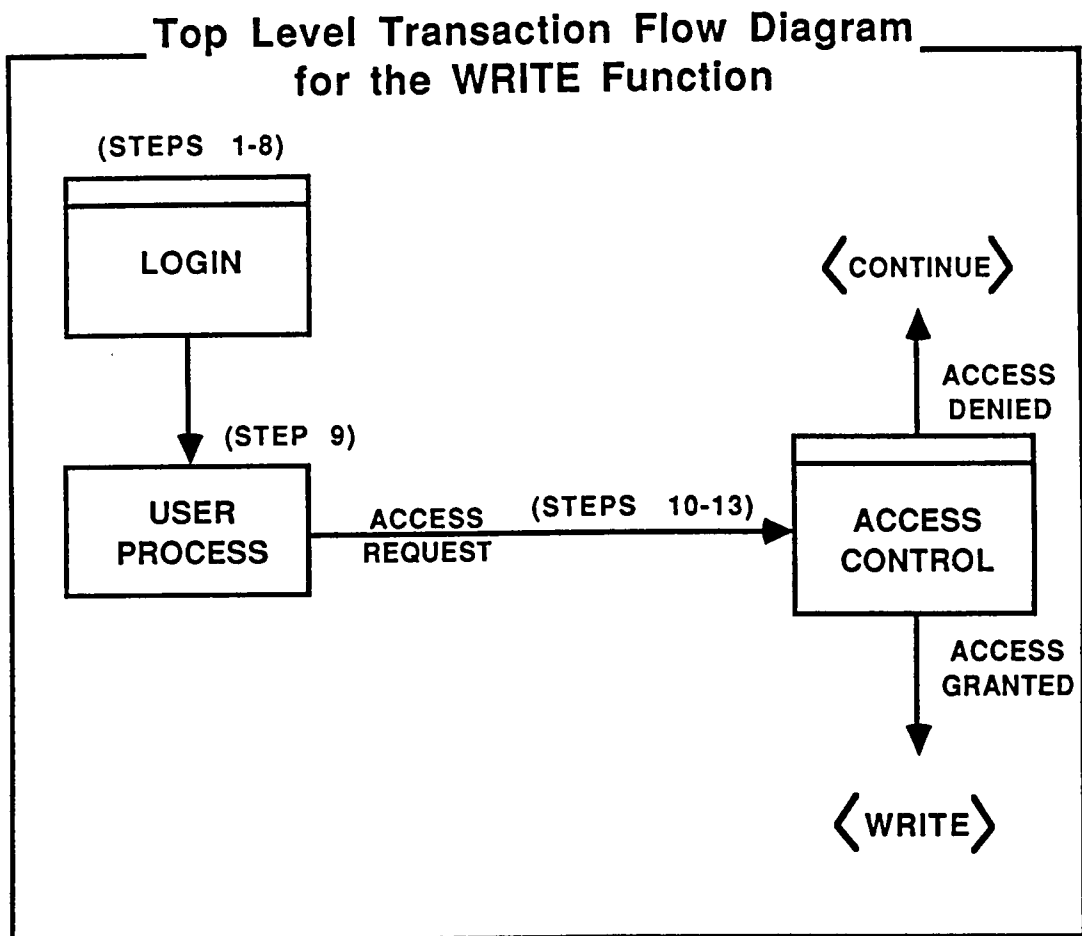


Fig. 4. Transaction Flow Diagram for the WRITE function.

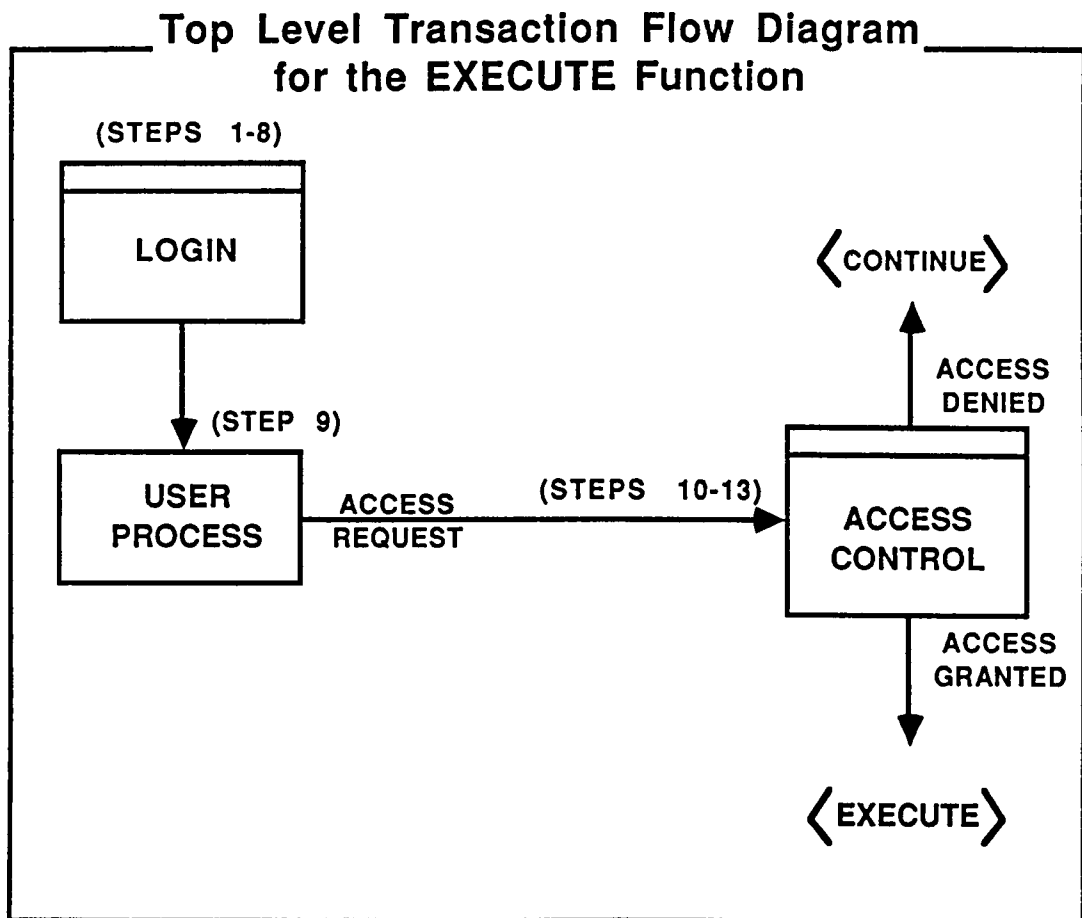


Fig. 5. Transaction Flow Diagram for the EXECUTE function.

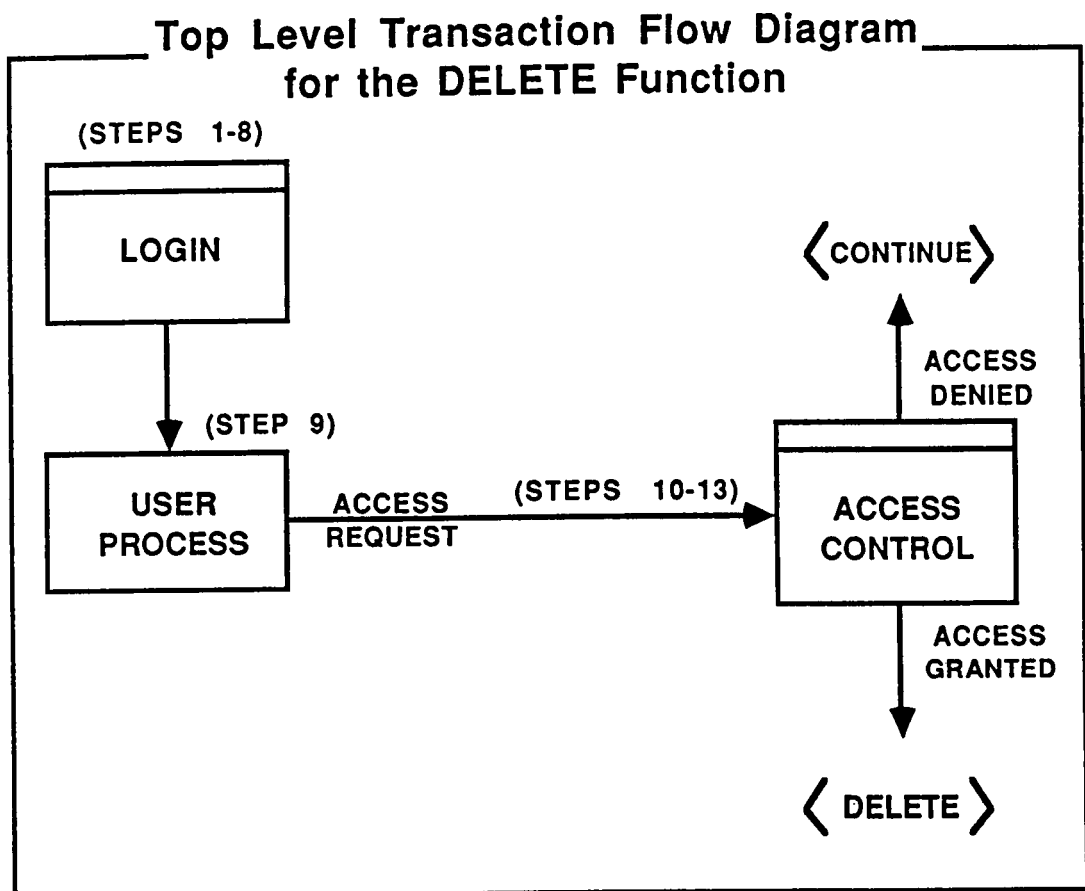


Fig. 6. Transaction Flow Diagram for the DELETE function.

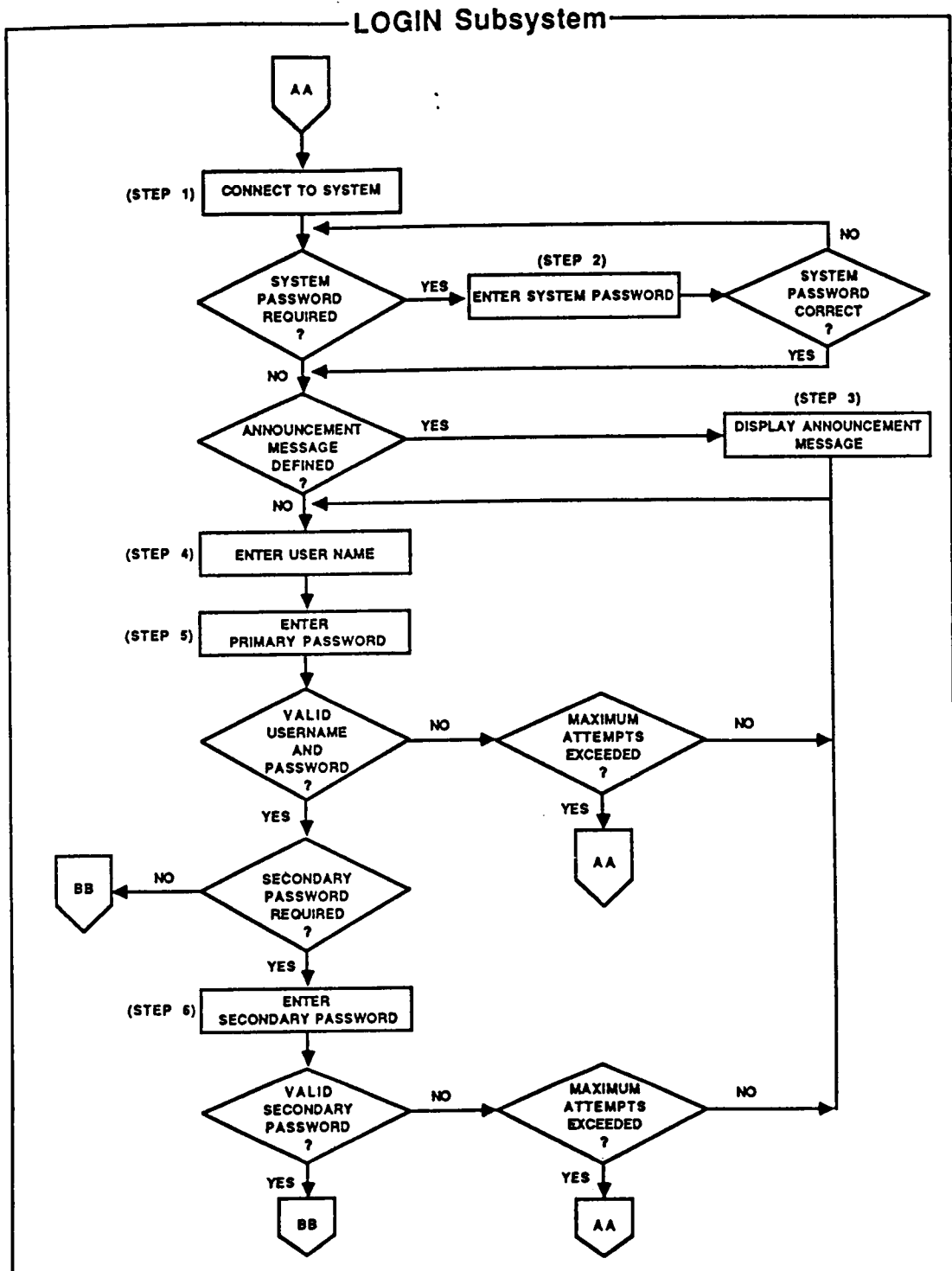


Fig. 7. Transaction Flow Diagram for the LOGIN Subsystem.

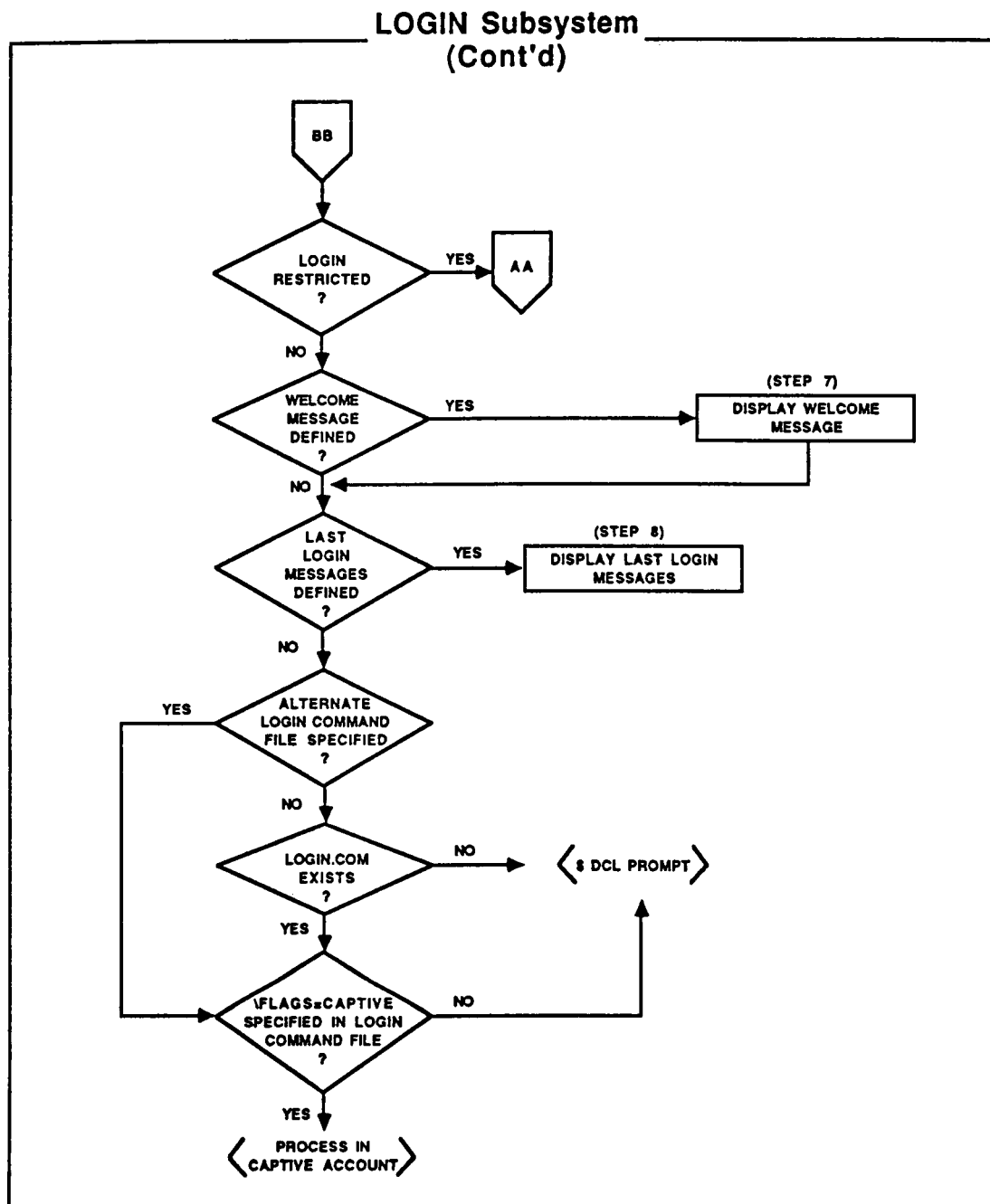


Fig. 7. (continued)

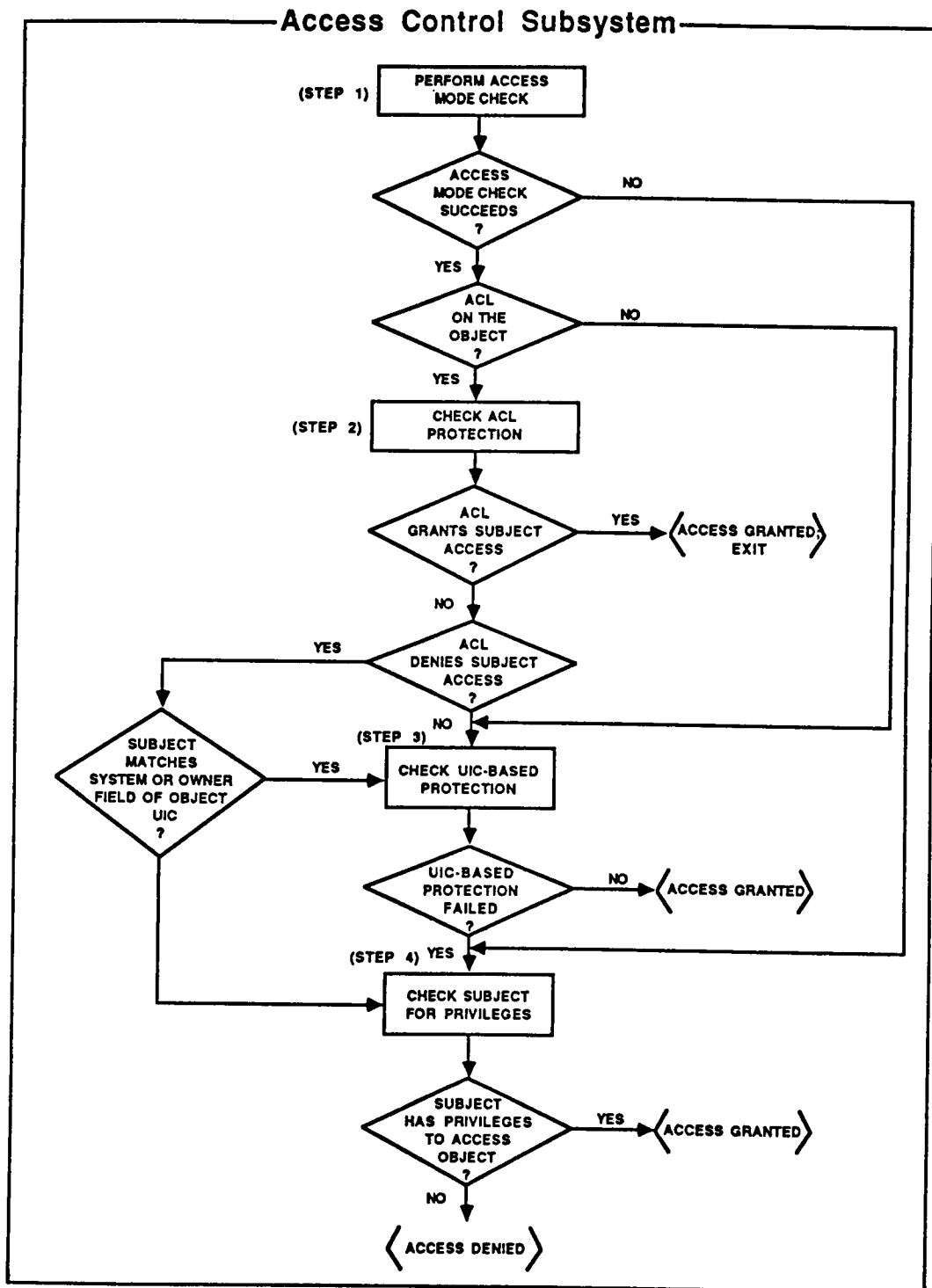


Fig. 8. Transaction Flow Diagram for the ACCESS CONTROL Subsystem.

3.11 EVALUATION OF TRANSACTION FLOWS FOR EXPOSURES

Each transaction flow diagram depicts the steps required to perform the transaction. Each step in the transaction flow diagram is evaluated to determine the potential exposures associated with that step and the set of controls provided to prevent exploitation of each exposure. The exposure/control sets mapped to each step of the transaction are shown in Appendix D.

3.12 EVALUATING CONTROL ADEQUACY

Based on the evaluation of the exposure/control sets associated with each step of each transaction, one must next determine if the controls provided in an operating environment satisfy the established system security requirements. For each identified exposure, the set of applicable controls are examined to determine if an exploitation of the exposure would be either prevented or detected by the associated controls.

Table 2 identifies the effectiveness of the controls listed for each exposure in Appendix C. The specific control that prevented exploitation of the exposure does not need to be identified; it is important only that the exposure be prevented by some control or combination of controls. Therefore, the effectiveness rating is for the class of controls that addresses the listed exposure, rather than for any one control.

The effectiveness of the control is given one of three ratings: 1. preventing the exposure; 2. detecting, but not preventing the exposure; or 3. neither detecting nor preventing the exposure.

Table 2. Control Effectiveness

EXPOSURE (See Appendix D)	Control Effectiveness		
	Prevent	Detect	Neither
E1	E1/P		
E2	E2/P		
E3	E3/P		
E4	E4/P		
E5	E5/P		
E6	E6/P		
E7	E7/P		
E8	E8/P		
E9	E9/P		
E10	E10/P		
E11		E11/D	
E12	E12/P		
E13	E13/P		
E14	E14/P		
E15	E15/P		
E16	E16/P		
E17	E17/P		
E18	E18/P		
E19	E19/P		
E20	E20/P		
E21			E21/N
E22	E22/P		
E23	E23/P		
E24	E24/P		
E25	E25/P		
E26	E26/P		
E27	E27/P		
E28	E28/P		
E29	E29/P		
E30	E30/P		
E31	E31/P		
E32	E32/P		
E33			E33/N
E34			E34/N
E35			E35/N
E36			E36/N
E37			E37/N
E38	E38/P		
E39	E39/P		
E40	E40/P		

Table 2. (continued)

EXPOSURE (See Appendix D)	Control Effectiveness		
	Prevent	Detect	Neither
E41	E41/P		
E42	E42/P		
E43	E43/P		
E44	E44/P		
E45	E45/P		
E46	E46/P		
E47	E47/P		
E48	E48/P		
E49	E49/P		
E50	E50/P		
E51	E51/P		
E52	E52/P		
E53	E53/P		
E54			E54/N
E55	E55/P		
E56	E56/P		
E57	E57/P		
E58			E58/N
E59	E59/P		
E60	E60/P		
E61	E61/P		
E62	E62/P		
E63	E63/P		
E64	E64/P		
E65	E65/P		
E66	E66/P		
E67	E67/P		
E68	E68/P		
E69	E69/P		
E70	E70/P		
E71	E71/P		
E72	E72/P		
E73	E73/P		

3.13 DATABASE

A relational database management system similar to the implementation developed by Pfleeger, Pfleeger, and Theofanos³⁹ for use with this methodology can be utilized to support the Basic Evaluation analysis. In the Basic Evaluation, there can be many exposures related to one requirement, one exposure related to many requirements, or one exposure related to one requirement. The relationships between exposures and controls are the same. Thus, the database must be capable of supporting one-to-one, one-to-many, or many-to-one relationships. The database must accommodate large text fields to handle the text descriptions needed for the requirements, exposures, controls, and transactions.

The contents of each table discussed in this analysis are listed in Table 3. A description of the table elements is listed after each table name.

Table 3. Database tables and elements.

TABLE	ELEMENTS
REQS	requirement ID, requirement description (text)
EXPS	exposure ID, exposure description (text)
CTRLS	control ID, control description (text)
TRANS	transaction ID, transaction description (text)
TRANSTEP	transaction ID, steps of each transaction
STEPS	step ID, step description (text)
REQEXP	requirement ID, exposures for each requirement
EXPCTRL	exposure ID, controls for each exposure, effectiveness

Table 4 lists the data elements used to link the tables. The names of the linked tables are followed by the name of the data field used to form the link. This table is called EXPCTRL.

Table 4. Links between database tables.

LINKED TABLES	LINKING ELEMENTS
TRANS - TRANSTEP	transaction id
TRANSTEP - STEPS	step id
EXPS - EXPCTRL	exposure id
EXPCTRL - CTRLS	control id
REQS - REQEXP	requirement id
REQEXP - EXPS	exposure id

Four tables are needed to represent the requirements, exposures, controls, and transactions in the system. The rows in these tables consist of a code identifying the requirement, exposure, control, or transaction and a text description. In this analysis the tables are called REQS, EXPS, CTRLS, and TRANS.

Another table is needed to detail the steps of the transactions. A row in this table should contain a transaction identification code (ID) and the steps for that transaction, which are represented with a step code derived from the transaction code concatenated with the step number. This table is called TRANSTEP.

The steps of each transaction are individually represented in another table called STEPS where each row consists of a step code and a text description of that transaction step. The step code identifies the transaction and the step number.

Another table called REQEXP associates each requirement with the applicable exposures. The columns of this table include a requirement code and the associated exposure codes. The mapping of controls to exposures is maintained with a similar table consisting of a column of exposure codes followed by the associated control codes. A column indicating the effectiveness of the controls in preventing or detecting exploitation of the exposure is also useful in this table.

Figure 9 illustrates the relationships between the database tables. With this database implementation a range of reports can be generated to facilitate the testing process. Test scripts to be used in the Detailed Evaluation can be automatically generated from the tables described.

DMBS11/2/89

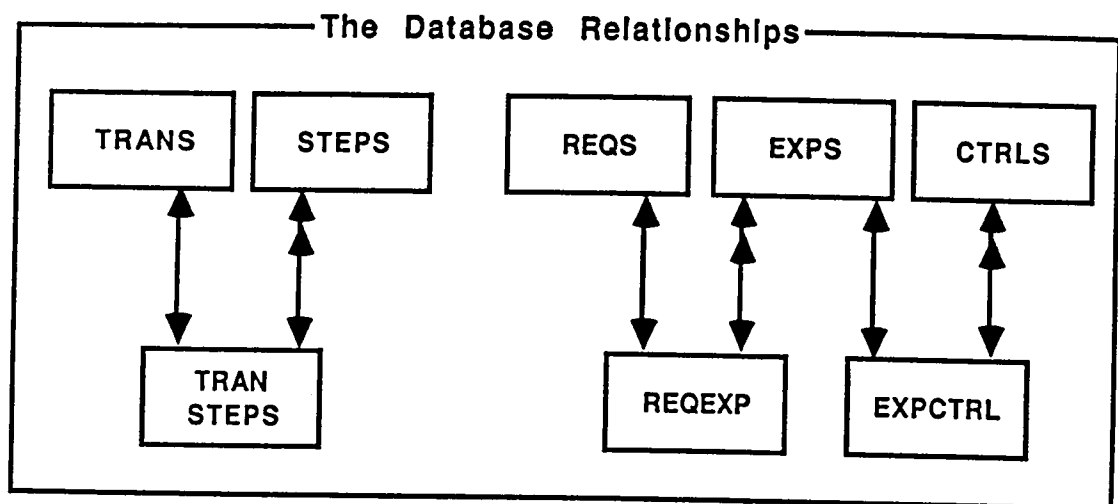


Fig. 9. The database relationships.

3.14 BASIC EVALUATION RESULTS

The requirements list compiled for the computer operating environment being evaluated was based on projected requirements expected to be mandated in the next few years. As a result, some of the requirements were not met in the Basic Evaluation. Although the requirement for mandatory access controls is planned for the future, the agency's computer operating environments do not implement labeling at this time.

VMS version 4.3 has been certified by the NCSC at the C2 level system⁴. Thus, it can be assumed that VMS provides the necessary control measures to satisfy the requirements for certification at that level.

With the exception of the labeling requirements, only one vulnerability was identified in the Basic Evaluation. If a terminal becomes disconnected or is powered off, the terminal session is not automatically terminated. This exposure could allow an unauthorized user to utilize that session's connection. However, there is a locally-implemented control that terminates an inactive process after a specified number of seconds. Thus, an unauthorized user would have to connect to the session before the process timed out. Controls were identified to satisfy all other security requirements.

4. DETAILED EVALUATION

4.1 DETAILED EVALUATION METHODOLOGY

The objective of the Detailed Evaluation is to test the effectiveness of the operating environment's controls in preventing exploitation of the exposures identified in the Basic Evaluation. Each exposure is tested in context with all associated transaction steps identified for each test user profile. Exhaustive application of this approach will provide comprehensive testing of all known instances where a user performing routine system functions could violate the system security requirements²⁵. Figure 10 depicts the Detailed Evaluation process.

4.2 OUTLINE OF STEPS REQUIRED TO PERFORM THE DETAILED EVALUATION

The steps of the Detailed Evaluation are:

1. For each test script produced in the Basic Evaluation, identify the transaction steps and the associated exposures and controls.
2. Perform testing activities to challenge the controls associated with each exposure for each step of each transaction, excluding redundant steps.
3. Prepare a report of test results.

Although this type of exhaustive testing of all cases could be impractical in environments with many types of transactions, the Detailed Evaluation can be focused on certain activities depending on the perceived importance of the known security exposures. This approach allows the system evaluator to determine the areas in need of intense testing and to concentrate the Detailed Evaluation on those areas of concern²⁵.

By eliminating redundant testing and allowing the system evaluator to tailor the testing for the specific environment, the Detailed Evaluation is reduced to a manageable task without compromising the completeness of the evaluation.

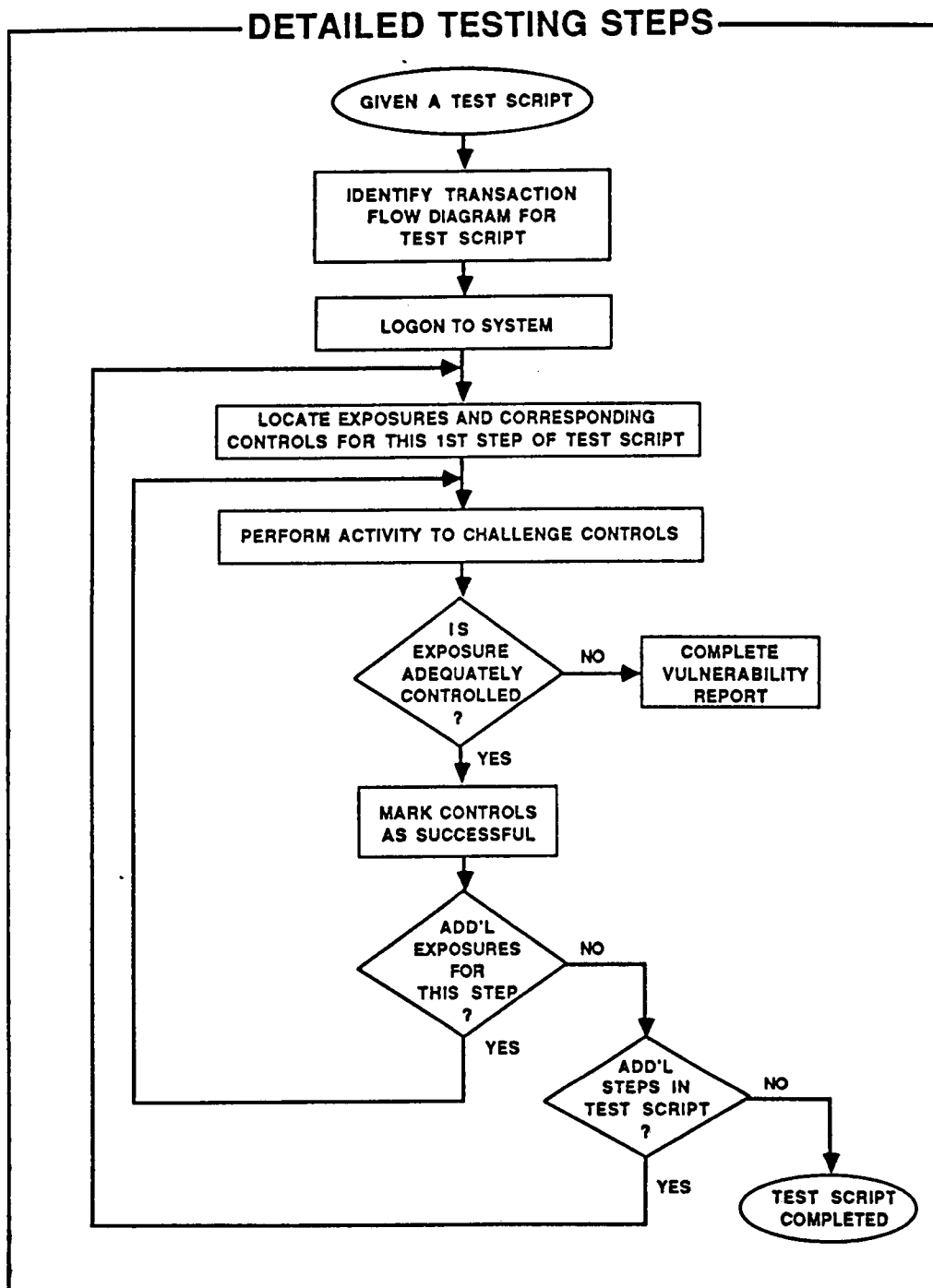


Fig. 10. The Detailed Evaluation process.

4.3 IDENTIFYING TEST SCRIPTS

Test scripts used to perform the Detailed Evaluation can be automatically generated from the database used to maintain the data determined from the Basic Evaluation. A test script is produced for each transaction; it consists of each step in the transaction, the exposures associated with each transaction step, and the controls provided to protect the exposure²⁵. The test scripts generated from the Basic Evaluation are included in Appendix D.

4.4 PERFORMING TESTING ACTIVITIES

Each exposure associated with a transaction step is challenged to determine whether the associated controls are effective in preventing exploitation of the exposure. The performance of the controls associated with each transaction step must be examined thoroughly. The effectiveness of the controls being tested could vary greatly with user profiles holding different privileges; therefore a comprehensive set of test user profiles is necessary for an effective evaluation. Each step of the transaction flow is repeated for each representative test user²⁵.

Testing of exposures for certain steps could be redundant in transactions that perform similar functions. For example, the READ and WRITE transactions defined in the Basic Evaluation are very similar, varying only in the type of access granted. Since the testing is the same for many steps of the transaction, it is not necessary to repeat testing for all steps in both transactions. Thorough testing of only one of the redundant steps is sufficient²⁵.

4.5 REPORTING TEST RESULTS

When vulnerabilities in the system are discovered, the system evaluator documents the test results in a vulnerability report. The report includes a description of the test that revealed the vulnerability, the conditions under which the test was performed, the expected test results, and the actual test results²⁵.

4.6 DETAILED EVALUATION RESULTS

The Detailed Evaluation for this government agency's operating environment need only include the optional VMS features that must be enabled to meet the C2 level requirements and the locally implemented controls. The inherent VMS controls that implement the requirements for a C2 level system are given. Furthermore, the agency administrative controls can only be verified; there is no test performed to confirm these controls.

An exhaustive Detailed Evaluation was not performed to develop the test methodology. However, many of the security features were tested for the unprivileged user. Some vulnerabilities were identified as a result of the detailed testing performed to expedite the development of the security evaluation methodology. Although VMS controls were identified in the Basic Evaluation to circumvent these exposures, the Detailed Evaluation revealed the existence of the following vulnerabilities:

1. The Announcement Message displayed system information identifying the system and the node before the user had successfully logged into the system.
2. The default file protection for one unclassified sensitive system allowed all access types to the SYSTEM user category. The default protection should not allow access to any user category other than OWNER on computers used for sensitive processing.
3. The default directory protection for one unclassified sensitive system allowed READ, WRITE, and EXECUTE access to the SYSTEM user category; READ and EXECUTE access to the GROUP user category; and EXECUTE access to the WORLD user category. The default protection should allow access to only the OWNER category on computers used for sensitive processing.

The Detailed Evaluation confirmed the vulnerability identified in the Basic Evaluation concerning the initialization of terminals when powered-on. The remaining features tested in the Detailed Evaluation were found to perform as expected from the Basic Evaluation.

5. CONCLUSIONS

5.1 OVERALL EVALUATION RESULTS

Most of the security requirements of the agency's computer operating environments were found to be satisfied. The major vulnerability identified in this security evaluation was the absence of mandatory access controls. The agency's current operating environments do not implement labeling at this time, but it is planned as a future requirement. Although VMS does not provide labeling capability, a layered product is available from Digital Equipment Corporation called the VMS Security Enhancement Service (VMS SES) that adds mandatory access controls to the VMS environment. Mandatory access control requirements could be met with this additional product. However, if this product or some other form of mandatory access controls is implemented, the security evaluation process would be conducted again to ensure proper performance of the new controls.

5.2 METHODOLOGY EFFECTIVENESS

This thesis presents the effective application of a proven security evaluation methodology to a computer operating environment³⁸. The concepts of transaction flow analysis were naturally adapted to consider the access control measures of the computer system. This approach to software security evaluations provides a practical, yet thorough, means of evaluating system security. A computer operating environment security evaluation for a stand-alone VAX/VMS system can be completed in approximately three months utilizing this methodology. However, the evaluation is only as effective as the requirements on which it is based. A complete and applicable set of system security requirements must be either provided or determined before the evaluation is performed. If the requirements are incomplete or ill-defined, then the evaluation process is without merit.

5.3 RECOMMENDATIONS

Some aspects of this evaluation methodology have the potential for automation. For instance, it is possible to implement a trusted routine to evaluate the system generation parameters and the system configuration files to determine the status of many of the optional operating system features, to determine default parameters, and to identify privileged users.

The concepts of this security evaluation methodology could be easily applied to many types of software systems. Such applications might include a database management system, an office automation system, or any of a broad range of software systems that can be viewed in terms of the flow of data within the system. Many of these systems are potential candidates for an adaptation of this evaluation methodology.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. *Department of Defense Trusted Computer System Evaluation Criteria*, Document 5200.28-STD, National Computer Security Center Department of Defense, December 1985.
2. *Evaluating Internal Controls in Computer-Based Systems, Audit Guide*, 0001-00113, U. S. General Accounting Office, June 1981.
3. *FAIS Early Operational Capability Software Evaluation Interim Report*, Joint Center for Information Security Technology, March 1988.
4. *Final Evaluation Report of Digital Equipment Corporation VAX/VMS Version 4.3*, CSC-EPL-SUM-86/004, National Computer Security Center, July 30 1986.
5. *Foreign Affairs Information System Early Operational Capability Security Requirements Allocation*, Planning Research Corporation, May 12 1986.
6. *Foreign Affairs Information System Early Operational Capability Software Certification Plan*, Joint Center for Information Security Technology, November 12 1987.
7. *Foreign Affairs Information System (FAIS) Early Operational Capability (EOC) System/Subsystem Specification*, Planning Research Corporation, McLean, Virginia, June 1986, under Contract 1043-606017.
8. *Foreign Affairs Information System Information Technology Plan, Version 2*, Planning Research Corporation, McLean, Virginia, August 1987, under Contract 1043-606017.
9. *Foreign Affairs Information System ISSO Security Manual*, Department of State, September 8 1987.
10. *Guideline for Computer Security Certification and Accreditation, Federal Information Processing Standards Publication 102*, U. S. Department of Commerce, National Bureau of Standards, September 1983.
11. *Guidelines for Security of Computer Applications*, FIPS Publication 73, June 1980.
12. *Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria*, NCSC-TG-005, Version-1, National Computer Security Center, July 31 1987.
13. P. Abbott, "Software Security in a VAX Environment," *University Computing (GB)* 9(3), 181 (September 1987).
14. C. R. Attanasio et al., "Penetrating an Operating System: A Study of VM/370 Integrity," *IBM Systems Journal* 15(1), 102-116 (1986).
15. Robert W. Baldwin, "Rule Based Analysis of Computer Security," pp. 227-233 in *COMPCON 92nd Digest of Papers. Intellectual Leverage*, IEEE Computer Society, IEEE Computer, Washington, DC, 1987.
16. D. E. Bell and E. L. Burke, *A Software Validation Technique for Certification: The Methodology*, MTR-2932-Vol-1, ESD-TR-75-54, MITRE Corporation, Bedford, MA, April 1975, under contract F19628-75-C-0001, AF-522B.

17. D. E. Bell and E. L. Burke, *A Software Validation Technique for Certification: The Methodology*, MTR 2932 VOL 1, ESD TR 75 54, MITRE Corporation, Bedford, MA, April 1975.
18. M. Bishop, *Analyzing the Security of an Existing Computing System*, NASA Technical Memo NASA-TM-89205, 1987.
19. M. Cheheyl et al., "Verifying Security," *Computing Surveys* 13(4), 279-339 (September 1981).
20. Robert A. Clyde, "VAX VMS Outlook," *DOD Trusted Computer Systems Criteria* 14(12), 208 209 (December 1985).
21. Digital Equipment Corporation, *Introduction to VMS System Services, VMS Version 5.0*, April 1988.
22. Digital Equipment Corporation, *VAX/VMS Version 5.0 System Management - Security*, April 1988.
23. Digital Equipment Corporation, *VMS System Manager's Manual, VMS Version 5.0*, April 1988.
24. Ronda R. Henning, Brian S. Hubbard and Swen A. Walker, "Computer Architectures, Database Security, and an Evaluation Metric," p. 518 525 in *Proceedings Third International Conference on Data Engineering*, IEEE Computer Society, IEEE Computer, Washington, DC, February 3 5, 1987.
25. S. E. Hicks, R. O. Chester and M. F. Theofanos, *Computer Security Generic Test Plan Description*, K/DSRD- -215, Martin Marietta Energy Systems, September 1989.
26. A. F. Huntley, *Computer Software Evaluation Methodology and Data Base Management System Selection*, ORNL 6260, Oak Ridge National Laboratory, Oak Ridge, Tennessee, April 1986, under contract AC05 84OR21400.
27. Lawrence J. Kenah, Ruth E. Goldenberg and Simon F. Bate, *VAX/VMS Version 4.4 Internals and Data Structures*, Digital Press, Bedford, Massachusetts 01730, 1988.
28. Harry T. Larson, "The Orange Book: The Department of Defense's Guide to Computer Security," *DOD Trusted Computer Systems Criteria* 6(10), 83(7) (October 1986).
29. R. Linde, "Operating System Penetration," pp. 361-368 in *Proceedings AFIPS 1975 National Computer Conference*, 1975.
30. Vin McLellan, "DoD Security Ratings: The Orange Book," *Digital Review* 3(19), 83 (1) (November 1986).
31. Gerald E. Murine and Jr. C. L. Carpenter, "Measuring Computer System Security Using Software Security Metrics," pp. 207-215 in *Computer Security: A Global Challenge, Proceedings of the Second IFIP International Conference on Computer Security, IFIP/Sec '84*, September 10-12, 1984.
32. William Neugent, "Acceptance Criteria for Computer Security," pp. 443-448 in *Proceedings AFIPS 1982 National Computer Conference*, 1982.
33. William Neugent, J. Gilligan, L. Hoffman and Z. Ruthberg, *Technology Assessment: Methods for Measuring the Level of Computer Security*, Special

- Publication 500-133, U. S. Department of Commerce National Bureau of Standards, 1985.
34. P. Neumann, *A Provably Secure Operating System: The System, Its Applications, and Proofs*, CS Lab Report CSL-116, SRI International, May 1980.
 35. "Computer System Security Evaluation," pp. 1087-1095 in *Proceedings of the 1978 National Computer Conference*, ed. Peter G. Neumann, AFIPS Press, 1978.
 36. G. H. Nibaldi, *Proposed Technical Evaluation Criteria for Trusted Computer System*, M79-225, MITRE Corporation, Bedford, MA, October 25, 1979, under contract F19628-80-C-001.
 37. T. A. Parker, "The Implementation of a Mandatory Security Policy in a Mainframe Operating System," p. 834 in *2nd European Conference on Computer Audit, Control and Security*, Dutch Assoc. Inf. (NGI), Amsterdam, Netherland, November 4-6, 1987.
 38. C. P. Pfleeger and S. L. Pfleeger, "A Transaction Flow Approach to Computer Security Certification for Document Handling Systems," *Computers & Security* 7, 495-502 (1988).
 39. C. P. Pfleeger, S. L. Pfleeger and M. F. Theofanos, *Data Base to Support a Transaction Flow Approach to Computer Security Certification*, K/DSRD-161, Martin Marietta Energy Systems, 1988.
 40. C. P. Pfleeger, S. L. Pfleeger and M. F. Theofanos, *A Methodology for Penetration Testing*, K/DSRD-107, Martin Marietta Energy Systems, March 31 1989.
 41. Charles P. Pfleeger, *Security In Computing*, Prentice Hall, Englewood Cliffs, NJ 07632, 1989.
 42. Zella G. Ruthberg and William Neugent, *Overview of Computer Security Certification and Accreditation*, Special Publication 500-109, U. S. Department of Commerce National Bureau of Standards, April 1984.
 43. E. T. Trotter and Peter S. Tasker, *Industry Trusted Computer System Evaluation Process*, MTR-3931, MITRE Corporation, Bedford, MA, May 1, 1980, under contract F19628-80-C-001.
 44. James A. Wall, "Investigation of Multilevel Security and Its Application in the Wargaming, Research, and Analysis (W.A.R.) Lab," Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1986.
 45. R. A. Whitehurst, "The Need for an Integrated Design, Implementation, Verification, and Testing Methodology," pp. 97-100 in *Proceedings of VERkshop III - A Formal Verification Workshop*, Department of Defense Computer Security Center, SIGSOFT Software Eng. Not., Watsonville, CA, February 18-21, 1985.

APPENDIXES

APPENDIX A
AGENCY REQUIREMENTS AND EXPOSURES

APPENDIX A: AGENCY REQUIREMENTS AND EXPOSURES

1. **Security Policy** - An explicit well-defined security policy must be enforced by the system. Given identified subjects and objects, there must be a set of rules that are used by the system to determine whether a given subject can be permitted to access a specific object.

- 1.1 **Discretionary Access Controls** - These controls are responsible for enforcing a need-to-know policy. This type of control allows access to certain objects to be restricted to those users with a formal need to know, regardless of user clearance.

1. The system shall provide discretionary access control mechanisms which shall define and control access between subjects and objects, thereby protecting objects from unauthorized access.

Exposures:

- Unauthorized subject allowed access to an object
- Authorized subject denied access to an object

2. Authorized users shall have the capability to define access rights to objects that they own.

Exposures:

- Unauthorized subject allowed access to an object
- Authorized subject denied access to an object

3. The discretionary access control mechanism shall provide operational security controls to limit propagation of access rights.

Exposures:

- Propagation of access rights to an object is unlimited, allowing uncontrolled granting of access permissions

4. The discretionary access control mechanism shall enforce a formal need to know policy in which the discretionary access control mechanism shall by default, provide that objects are protected from unauthorized access.

Exposures:

- Access control default allows access of objects to subjects without the need to know

5. File access permissions shall default to *no access* except for the owner of a file for classified and sensitive computers. On nonsensitive computers file access permissions shall default to READ and EXECUTE access for all users.

Exposures:

- Unauthorized subject allowed access to an object
- Authorized subject denied access to an object

6. Directory access permissions shall default to *no access* except for the directory owner for classified and sensitive computers. On nonsensitive

computers directory access permissions shall default to READ and EXECUTE access for all users.

Exposures:

- Unauthorized subject allowed access to an object
- Authorized subject denied access to an object

- 1.2 Object Reuse** - The operating system shall ensure that residue from user programs in both batch and on-line modes are cleared or written over before memory and other storage are allocated by the system for use by other user programs.

Exposures:

- Subject space in main memory not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject
- Subject space on secondary storage not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject
- Sensitive or classified information channeled to unauthorized subject

- 1.3 Labels** - Access control labels must be associated with objects in the system. In order to control access to information stored in a computer system it must be possible to mark every object with a label that reliably identifies the object's sensitivity level, and/or the modes of access accorded those subjects who may potentially access the object.

1. The system must maintain reliable (assigned or changed only by an authorized security administrator) labels on all objects to reflect the degree of protection required for the object and to reflect the authorization of a subject (user or processes) to access the object.

Exposures:

- Unauthorized subject modifies the classification level or sensitivity label of an object
 - Authorized subject not allowed to modify the classification level or sensitivity label of an object
 - Unauthorized subject allowed access to an object
 - Authorized subject denied access to an object
2. The system shall be capable of associating security labels with user terminals and other user-employable devices before allowing use of system resources.

Exposures:

- User accesses system from unauthorized device
 - User denied access to system from authorized device
3. A change in the classification level or sensitivity label of an object shall be controlled so that such a change can only be made by an authorized system security administrator.

Exposures:

- Unauthorized subject modifies the classification level or sensitivity label of an object
 - Authorized subject not allowed to modify the classification level or sensitivity label of an object
4. Labels shall be associated with electronic information being exported.

Exposures:

- Exported object is exported without the appropriate sensitivity label
5. Human readable (text) labels shall be associated with human readable output.

Exposures:

- Hardcopy output is produced without readable labels identifying the sensitivity of the information
6. Security related software shall request and receive the security level of data being input to the system.

Exposures:

- Data is entered into the system with no, or an incorrect, associated sensitivity label

1.4 Mandatory Access Controls (Clearance- Sensitivity Matching) - There must exist a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity.

1. The system shall enforce mandatory access control policies over all subjects and objects under its control.

Exposures:

- Unauthorized subject allowed access to an object
 - Authorized subject denied access to an object
2. The mandatory access controls shall include access controls based on the classification level of the data and the security clearance level of the user, i.e., no individual shall have access to classified data for which that person does not have the appropriate clearance.

Exposures:

- Unauthorized subject allowed access to an object
 - Authorized subject denied access to an object
3. Subjects and objects shall be assigned sensitivity labels that are a combination of hierarchical classification levels and non-hierarchical categories, and the labels shall be used as a basis for mandatory access control decisions.

Exposures:

- Unauthorized subject allowed access to an object

— Authorized subject denied access to an object

4. The mandatory access controls shall be supported by a mechanism that ensures the integrity of all data in the system, including labels that represent the sensitivity levels.

Exposures:

— Unauthorized subject modifies the classification level or the sensitivity label of an object

5. All users shall have a predefined user profile which shall determine their need to access objects. This user profile shall determine access right to all objects in the system.

Exposures:

— Unauthorized subject allowed access to an object

— Authorized subject denied access to an object

6. The system shall ensure that user profiles can be established and changed by only the system security administrator.

Exposures:

— Unauthorized subject creates or modifies user profiles

— Authorized subject not allowed to create or modify user profiles

2. **Accountability** - In order to mediate access control, there must exist a means to uniquely identify each individual subject and the objects that subject is to be allowed to access.

2.1 Identification and Authentication

1. The system shall require users to identify themselves before revealing any system information or allowing any access to system resources.

Exposures:

— Nonuser allowed access to system information

— System cannot associate the user with a user profile

— User accesses system from unauthorized device

— User denied access to system from authorized device

2. The system shall be able to enforce individual accountability by providing the capability to uniquely identify each individual user.

Exposures:

— System cannot associate user with a user profile

— System cannot associate an individual user with audited events

3. All users shall require individual user IDs and nondisplayable passwords.

Exposures:

— Nonuser allowed access to system

— System cannot authenticate user identity

- System cannot associate user with a user profile
 - System can not associate individual user with audited events
4. Passwords shall not be displayed at login.

Exposures:

- Passwords displayed to unauthorized or nonuser
5. The system shall protect authentication data including passwords so that it cannot be accessed or changed by any unauthorized subject.

Exposures:

- Unauthorized subject modifies authentication data
 - Authorized subject not allowed to modify authentication data
6. Authentication data used to verify the identity of individual users (e.g. passwords) shall not be stored in plain text within the computer system.

Exposures:

- Plaintext authentication data is stored in the computer system and exposed to unauthorized or nonuser
7. The authentication of users shall employ a one-way encryption technique for passwords.

Exposures:

- Plaintext authentication data is stored in the computer system and exposed to unauthorized or nonuser
 - Encrypted passwords are exposed to an unauthorized user who is able to decrypt them
8. Initial passwords shall be assigned and distributed to users of the system.

Exposures:

- Users with the ability to choose their own passwords choose simplistic passwords that are easily guessed by an unauthorized or nonuser
9. If users are allowed to change their passwords, then they are required to use the password generation utility to generate the new password.

Exposures:

- Users with the ability to choose their own passwords choose simplistic passwords that are easily guessed by an unauthorized or nonuser
10. Processes shall be identifiable to the host processor for control of I/O and for disconnection when not in use.

Exposures:

- Nonuser allowed access to system
 - Sensitive information channeled to unauthorized or nonuser
11. The time and date of the last successful login will be displayed at login.

Exposures:

— User is unaware of unauthorized access to his or her user account

2.2 Audit - Selective audit information pertaining to the security aspects of the system must be reliably maintained in order to trace security violations to the responsible party.

1. The system shall ensure the capability of associating the individual user identifier with all auditable actions taken by that individual.

Exposures:

— System can not associate an individual user with audited events

2. For each recorded event, the audit record shall identify:
 - a. Date and time of event
 - b. Success or failure of event
 - c. Type of event
 - d. User or process identification
 - e. For identification/authentication events the origin of the request (e.g., terminal ID) shall be included in the audit record.

Exposures:

— System fails to record sufficient information to detect attempted security violations

3. For events that introduce an object into a user's address space and for object deletion events, the audit record shall include the name of the object and the object's security level.

Exposures:

— Unauthorized access to an object is not logged

4. At a minimum, the following events shall be audited:
 - a. All attempts to access system resources, whether or not the attempt is successful
 - b. System faults and restarts
 - c. Actions of system security administrators and computer operators
 - d. User inactivity timeouts (thresholds shall be adjustable by the system security administrator)
 - e. Modification to user's authentication profile
 - f. Use of identification and authentication mechanisms (all logins)
 - g. Introduction of objects to subjects (file open, file close)
 - i. Creation and deletion of objects
 - h. User logoff activity

Exposures:

- System fails to audit minimum set of identified areas of potential security violations

5. A selective auditing capability shall be provided with the capability of the system security administrator to select the events to be audited.

Exposures:

- Necessary auditing capability not provided

6. The system shall provide protection to the events logs and audit files to prevent changes or deletions.

Exposures:

- Unauthorized subject modifies audit log to cover security violations
- Authorized security administrator modifies audit log to cover security violations

7. The system shall provide summary reports of user profiles and changes to the system security administrator.

Exposures:

- System security administrator is unaware of unauthorized creation or modification of user profiles

3. **Assurance** - In order to assure that the defined requirements are enforced by a computer system, there must be some identified and unified collection of hardware and software controls that perform those functions. These mechanisms are typically embedded in the operating system and are designed to carry out the assigned tasks in a secure manner.

- 3.1 **Operational Assurance** - The trusted mechanisms that enforce the basic requirements of the system must be continuously protected against tampering and/or other unauthorized changes.

1. The operating system shall be maintained in a domain for its own execution that protects it from external interference or tampering (e.g. by modification of its code or data structures).

Exposures:

- Unauthorized subject modifies operating system
- Authorized subject not allowed to modify operating system

2. The operating system shall enforce the READ, WRITE, EXECUTE, and DELETE privileges of a user with respect to any given file.

Exposures:

- Unauthorized subject allowed access to an object
- Authorized subject denied access to an object

3. Message stream modification countermeasures shall be used to deter and detect changes to sensitivity labels associated with data transported through the system.

Exposures:

- Sensitivity labels associated with classified objects are corrupted, allowing access by unauthorized subjects or access denial to authorized subjects
4. The operating system shall control the continued separation of user and privileged modes so that a program operating in user mode is prevented from unauthorized use of privileged functions.

Exposures:

- User mode program accesses privileged functions
5. System hardware and software features shall exercise control over the memory locations to which a user program has access.

Exposures:

- User program accesses main memory space of another user or the system
 - User program accesses auxiliary memory space of another user or the system
6. The operating system shall restrict memory dumps so that individuals are exposed only to the information or programs in memory for which they have access authority.

Exposures:

- Exposure of sensitive or classified information to unauthorized users by dumping areas of memory, including memory assigned to another subject
 - Programmer obtains access to data or files of another subject on secondary memory
7. One class of instructions reserved for exclusive use of the operating system and one class usable by both the operating system and the user's programs shall be provided.

Exposures:

- Unauthorized subject executes privileged instructions
 - Authorized subject not allowed to execute privileged instructions
8. A set of privileged instructions shall be provided which enable the operating system to restrict access of users to input/output devices, memory management functions, and all information which defines the execution characteristics of user processes.

Exposures:

- User mode program acquires access to input/output device, memory management function, user process executing characteristics
9. The hardware/software shall be designed so that all machine instructions/operation codes, both privileged and user, produce results in a predefined set of responses.

Exposures:

- Execution of a command produces unpredictable, potentially detrimental, results

10. Error detection shall be performed on each FETCH cycle in order to avoid erroneous commands, data and/or addressing which may result in unauthorized or incorrect action by the computer.

Exposures:

- Execution of a command produces unpredictable, potentially detrimental, results

11. The operating system alone shall control input/output operations, memory management functions, and maintenance of execution state information.

Exposures:

- User mode program acquires access to input/output device, memory management function, user process executing characteristics

12. Registers shall be protected by error detection or redundancy checks to provide secure operating of the hardware.

Exposures:

- Execution of a command produces unpredictable, potentially detrimental, results

13. Initial login and authentication shall be affected via a direct communication path between the user login input device and the security related software of the system.

Exposures:

- Programmer inserts program that masquerades as security program and obtains user ID and password of another user

14. Login attempts by any user shall be limited. The number of attempts allowed is a system parameter under the control of the system administrator.

Exposures:

- Nonuser allowed unlimited trial-and-error attempts to login to the system

15. If the number of login attempts exceeds the number allowed, the user shall be locked out by the system. System reinitialization shall not reinitialize the lockout state of any user.

Exposures:

- Nonuser allowed unlimited trial-and-error attempts to login to the system
- Repeated unsuccessful login attempts are not noticed by the system security administrator

16. User authorization after lockout shall be reinitiated by a system administrator.

Exposures:

- Repeated unsuccessful login attempts are not noticed by the system security administrator
17. Interactive processes shall be automatically terminated after “n” minutes of inactivity. The timeout interval shall be flexible as defined by the system security administrator.

Exposures:

- Automatic timeout of process is disabled
 - User leaves logged-on terminal unattended for extended periods of time
18. The capability to selectively lockout specific terminals on the system by the system administrator shall be provided.

Exposures:

- User terminal cannot be locked out by the system administrator
19. The system shall be able to deny or restrict users to certain system resources (workstations, processors, tape drives, etc.).

Exposures:

- Unauthorized subject accesses restricted system resources
 - Authorized subject denied access to restricted system resources
 - User accesses system from unauthorized device
 - User denied access to system from authorized device
20. A power-up of the system shall result in reinitializing all terminals except those locked out by the system administrator. Also, the power-up of a terminal shall reinitialize the terminal except those locked out by the system administrator.

Exposures:

- Power up of a terminal preserves pre-existing user session
21. 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 system.

Exposures:

- Execution of a command produces unpredictable, potentially detrimental, results

3.2 Life-Cycle Assurance - The actual development of the operating system is controlled by the product vendor. Adherence to any life-cycle requirements before installation in an agency operating environment can only be confirmed by the agency security personnel.

1. Security Testing — The security mechanisms of the system shall be tested and found to work as claimed in the system documentation prior to production use.

Exposures:

— Non-obvious security exposures exist in the system

2. Configuration Management — Any local modifications made to the operating system code shall be controlled by a configuration management system.

Exposures:

— Uncontrolled modification results in inconsistency among components of the system

— Introduction of security exposures due to object modification

4. **Documentation** - The basis for trusting the system control mechanisms in their operational setting must be clearly documented such that it is possible to independently examine the evidence to evaluate their sufficiency.

4.1 Security Features User's Guide

1. A single summary, chapter or manual in user documentation shall be maintained that describes the protection mechanisms provided by the system, guidelines on their use, and how they interact with one another shall be provided.

Exposures:

— Misuse or nonuse of security features due to lack of functional understanding

4.2 System Security Administrator's Manual

1. A system security manual addressed to the system security administrator shall be maintained that presents cautions about functions and privileges that should be controlled when running a secure system.

Exposures:

— Misuse or nonuse of security features due to lack of functional understanding

2. The system security manual shall also describe the procedure for configuring the automated security functions of the system and other administrative procedures for system security.

Exposures:

— Misuse or nonuse of security features due to lack of functional understanding

3. 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.

Exposures:

— Misuse or nonuse of audit capabilities due to lack of functional understanding

4. The manual shall describe the operator and administrator functions related to security, to include changing the security characteristics of a user.

Exposures:

- Misuse or nonuse of security features due to lack of functional understanding
5. The manual shall provide guidelines on the consistent and effective use of the protection features of the system, how they interact, how to generate securely a new system and procedures, warnings and privileges that need to be controlled in order to operate the system in a secure manner.

Exposures:

- Misuse or nonuse of security features due to lack of functional understanding
6. Configuration policies and procedures concerning system administration shall be described.

Exposures:

- Misuse or nonuse of security features due to lack of functional understanding

4.3 Test Documentation

1. A document shall be provided that describes the system test plan, test procedures that show how the security mechanisms were tested, and results of the security mechanisms' functional testing.

Exposures:

- Software has not been sufficiently tested by the proper agencies to ensure that security exposures do not exist

4.4 Design Documentation

1. A document shall be provided that describes the designer's philosophy of protection and an explanation of how this philosophy is translated into the system.

Exposures:

- Security exposures exist due to lack of comprehensive consideration of all possible security exposures
2. The interfaces between the system modules shall be described.

Exposures:

- Security exposures exist due lack of understanding of the interface between system modules

5. Administrative Requirements - Additional administrative requirements are necessary to encompass a comprehensive security policy.

1. Procurement shall be limited to those companies who are reputable, bonded, and have signed certification statements declaring their products to be free of subversive code (trojan horses, time bombs, viruses, worm-hole routines, etc.) Where local modifications are required, source code shall be furnished or the appropriate hooks provided.

Exposures:

- Software has not been sufficiently tested by the proper agencies to ensure that security exposures do not exist.
- 2. An ADP Security Plan shall be submitted to the proper accrediting official, the Computer Security Operations Manager (CSOM), for approval before classified information can be processed on a computer system. Further, if the system is a network of components not under the jurisdiction of the same CSOM the accrediting official is the Computer Security Program Manager (CSPM). This effort must be coordinated with the site computer security group.

Exposures:

- Computer system used for processing classified data has not been approved for processing of classified data
- Unauthorized subject is allowed access to an object
- 3. A Computer Protection Plan shall be submitted by the Computer Protection Program Manager (CPPM) to the appropriate site management for approval before unclassified sensitive information can be processed on a computer system.

Exposures:

- Computer system used for processing unclassified sensitive data has not been approved for processing unclassified sensitive data
- 4. Periodic risk assessment reviews shall be conducted in accordance with agency orders for unclassified sensitive computers and classified computers.

Exposures:

- Changes to the system or the physical environment produce new exposures that have not been considered in the current security plan
- 5. Backup procedures shall be implemented as appropriate to ensure minimal loss of data in the event that data restoration from backup media is necessary. One copy of the data backups shall be kept in a different location.

Exposures:

- Data is lost due to the restoration of backup data
- Data backups are not maintained, resulting in excessive data loss in the event of media damage
- 6. A contingency recovery plan shall be developed and periodically reviewed to ensure that the plan is kept current.

Exposures:

- Some event (i.e., natural disaster) yields the system inoperable, and there is no means to continue processing
- 7. All vendor supplied or publicized accounts shall be disabled or the passwords changed.

Exposures:

- Unauthorized user is allowed access to an object
8. The number of users having access to privileged accounts shall be held to a minimum. Up-to-date lists of those persons having access to privileged accounts and the accounts themselves shall be submitted to the appropriate security authority. This list shall include users with the ability to grant privileges.

Exposures:

- User access to privileged accounts is not monitored, thus users have access to privileged accounts without an established need
9. There shall be no non-encrypted dial-up access or non-encrypted dial-out capability for classified computers.

Exposures:

- Nonuser allowed access to system
 - User accesses system from unauthorized device
 - User denied access to system from authorized device
 - Sensitive information channeled to unauthorized or nonuser
10. There shall be no proxy access to privileged accounts on the system.

Exposures:

- Unauthorized user gains access to a privileged account and assumes system control
11. For all dial-up connections, the session shall automatically be terminated in the event a terminal is disconnected. This prevents another user from connecting to the disconnected session.

Exposures:

- Dial-up terminal becomes disconnected, leaving the session open for any user who connects to that port
12. All dial-in/dial-out capability shall be under the control of a central management of dial-up access. [Note: This is an installation specific requirement in some agency installations.]

Exposures:

- Dial-up access is not monitored, resulting in unknown external access to the system

APPENDIX B
AGENCY CONTROLS

APPENDIX B: AGENCY CONTROLS

- C1 VMS provides two types of logins: interactive or noninteractive. Interactive logins are performed in a series of steps where the system prompts for information and the user provides it. Noninteractive logins are performed by the system and require no user interaction. VMS logins are further differentiated into the following classes:

local	interactive
dialup	interactive
remote	interactive
network	noninteractive
batch	noninteractive
detached	dependent on parent process
subprocess	noninteractive

- C2 Logins can be restricted to include or to exclude any of the following classes: LOCAL, REMOTE, DIALUP, BATCH, or NETWORK. The general name INTERACTIVE can be used to include or to exclude all three of the classes LOCAL, REMOTE, and DIALUP using one expression.
- C3 A limited amount of time is allotted to complete the login process.
- C4 Logins can be restricted to certain times of the day and certain days of the week. These restrictions are imposed on classes of logins.
- C5 Proxy logins represent a special type of network login. With a proxy login, you can gain network access without including an access control containing the user name and password in the network login request.
- C6 Four types of accounts are provided by VMS systems:
1. Open accounts which require no password; the password is null.
 2. Captive accounts which may require a password; these limit user operations.
 3. Accounts that require passwords and prohibit the user from changing the password.
 4. Accounts that require passwords, but the user has the capability to change the password.
- C7 An expiration date can be specified for an account by the system security administrator.
- C8 Users must identify themselves with a user name and password to access the system.
- C9 Passwords are not displayed during login.
- C10 A secondary password can be required in addition to the primary password for an account.
- C11 System passwords can be required to gain access to a terminal.
- C12 A one-way encryption algorithm is applied to passwords before they are stored in the system.

- C13 The system security administrator can control the number of opportunities a user is allowed to enter a correct password during a dialup login before the connection is automatically broken.
- C14 The system manager can require that users use the automatic password generator when changing their password or can allow the user to choose the method of changing their password.
- C15 The system security administrator can require a minimum length for passwords.
- C16 The system security administrator can specify a maximum length of time that a user can retain the same password.
- C17 Break-in evasion measures may be enabled by the system security administrator. These measures are triggered when the maximum number of unsuccessful password attempts is exceeded over a predetermined period. The maximum number of unsuccessful attempts and the time period are determined by the system security administrator. When break-in evasion is triggered, all logins for the suspect user name are disabled for a specified interval.
- C18 The announcement message typically identifies the node being accessed and immediately precedes the *Username* prompt. The content of this message can be controlled or disabled by the system security administrator.
- C19 The welcome message is displayed after a successful login and usually displays the VMS version number and possibly the node name. The welcome message can be changed or suppressed entirely by the system security administrator.
- C20 A group of three messages providing information about the last successful login can be displayed just after the welcome message. These last login messages are optional and are enabled or disabled as a group. If the messages are enabled, from one to three of the following messages are displayed:
 1. Last successful interactive login message - provides the time of the last completed login for a local, dialup, or remote login.
 2. Last successful noninteractive login message - provides the time of the last completed login for a batch or network login.
 3. Number of login failures - displays the number of unsuccessful login attempts that have occurred due to an incorrect password since the last successful login.
- C21 A disconnected job message can be displayed to indicate that a previous login was prematurely interrupted but is available for reconnection. Two conditions must exist for this message to appear:
 1. The terminal where the interruption occurred must have been set up as a virtual terminal to prohibit a process from being disconnected when the line sensed a hangup.
 2. During a recent session for the user now logging on, the connection to the CPU through that terminal must have been broken.
- C22 A secure terminal server feature can be enabled for a terminal to ensure that the VMS login program is the only program able to receive a login.
- C23 VMS provides a file called a rights database that contains a list of special names called identifiers as well as a list of users specified as holders of identifiers. When

a user logs in the identifiers that the user holds in the rights database (including the UIC and the system- defined identifiers) are copied into a rights list that is part of the user process. This rights list is the structure that VMS uses to perform all protection checks.

- C24** VMS provides two primary means of access control: UIC-based protection and ACL protection. Access to an object is determined from an interaction between ACLs, the UIC-based protection, and the user privileges.

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

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

- C25** A standard UIC-based protection based is applied to all user files. The UIC-based protection controls access to objects according to the user categories SYSTEM, OWNER, GROUP, and WORLD. The VMS system uses the object's protection code to determine access to an object for each category. Each category of user can be allowed or denied any of the following types of access through the protection code: READ, WRITE, EXECUTE, or DELETE. VMS provides UIC-based protection for the following objects: disk files, directories, volumes, global sections, devices, logical name tables, and queues.

Each user is assigned a UIC by the system manager via the VMS Authorize Utility. The UIC for each user is defined in the system user authorization file (UAF). Each system object has an associated UIC, defined to be the UIC of its owner, and a protection code that defines who is allowed what type of access. The relationship between the UIC of the user and the UIC of the object controls access to the object.

The UIC is translated into a 32-bit value that represents a group number and a member number. The resulting 32-bit numeric UIC is kept in the system rights database, which is a file containing information pertaining to the access rights of the subject.

The method of storing alphanumeric UICs dictates that member names must be unique, and that no member can participate in more than one group.

- C26** ACL protection is an optional form of protection available for the following types of objects: devices, disk files, directories, group global sections, system global sections, logical name tables, and queues. ACLs can be used to grant or deny file access to individual users or groups of users, independent of the UIC.

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

1. UIC identifiers - depend UICs which uniquely identify each user on the system.
2. General identifiers - defined by the security manager in the system rights database to identify groups of users on the system.
3. System-defined identifiers - describe certain types of users based on their use of the system. For example, BATCH, NETWORK, DIALUP, INTERACTIVE, LOCAL, and REMOTE would correspond directly to the type of login the user executed.

Each ACL consists of one or more ACE that grants or denies access to a particular system object. The following are the three types of ACEs involved with security:

1. Identifier ACE - controls the type of access allowed to a particular user or group of users.
2. Default Protection ACE - defines the default protection for a directory so protection can be propagated to the files and subdirectories created in that directory.
3. Security Alarm ACE - provides an alarm message when an object is accessed to alert managers to possible security threats.

ACLs can be created by the system by default, by the system security administrator for specific objects, and by users to protect their own files.

C27 The VMS system features privileges that the system and security managers can assign to users when they create or modify user accounts.

Four system privileges affect user access, regardless of the access dictated by either an ACL for the object or the access granted through matching the user's category in the protection code. These privileges are: SYSPRV, GRPPRV, READALL, and BYPASS.

1. SYSPRV - A user with SYSPRV privilege receives the access accorded to users in the SYSTEM category.
2. GRPPRV - A user with GRPPRV privilege whose UIC group matches the group of the owner of the object receives the same access accorded to users in the system category. Thus, the user with GRPPRV privilege is able to manage a group's files.
4. READALL - A user with READALL privilege receives READ and CONTROL access to the object, even if that access is denied by the ACL or UIC-based protection. In addition, the user may receive any other access granted through the protection code.
3. BYPASS - A user with BYPASS privilege receives all types of access to the object, regardless of its protection.

A user cannot execute an image that requires a privilege the user does not possess unless the image is installed as a known image with the privilege in question.

C28 Control access is a type of access that can be specified in an ACL and is automatically granted to users with certain user categories when UIC-based protection is evaluated. CONTROL access grants the accessor all the privileges of the object's actual owner.

- C29** Attributes are identifier characteristics that can be specified when adding identifiers to the rights database or granting identifiers to users. The attributes are:
1. Resource - allows holders of the identifier to charge disk space to the identifier.
 2. Dynamic - allows holders of the identifier to remove the identifier from the process rights list using the DCL command SET RIGHTS_LIST.
- C30** VAX hardware architecture provides four processor access modes: user, supervisor, executive, and kernel. These access modes are used to determine which instructions a process is allowed to execute and which memory locations those instructions can access. Thus, the access modes protect the VMS system from processes executing in user mode as well as separate parts of the operating system according to their degree of privilege.
- C31** The VMS memory management architecture protects the process address space of one process from code running in the context of another process. Although system virtual address space is addressable by all processes, page-by-page protection controls access to specific virtual pages for certain access modes.
- The memory management system protects the code and data used by more privileged access modes from modification by less privileged modes.
- C32** The VMS operating system controls allocation of its system-wide resources, such as nonpaged dynamic memory and page file space, through the use of quotas and limits. These process attributes are assigned when the process is created. By restricting these system resources, VMS exercises control over the resource drain that can be exerted by a single process.
- C33** Disk volumes are provided protection at the file, directory, and volume levels.
- C34** Magnetic tape volumes are provided protection only at the volume level.
- C35** Record-oriented devices such as terminals, line printers, mailboxes, and special-purpose devices are provided protection.
- C36** Each account has an established default protection code for all files the user creates in the initial top-level directory.
- C37** Only users with ownership privileges to an object can change object ownership. A user must satisfy only one of the following conditions to have ownership privileges:
1. Hold the resource attribute to the identifier that owns the file.
 2. Qualify as a member of the SYSTEM user category, hold SYSPRV or BYPASS privilege, or hold a UIC that matches that of the owner of the volume containing the file or directory.
 3. Hold the GRPPRV privilege while also holding a UIC in the same group as the object owner.
- C38** Default ownership for a file is determined as follows:
1. An attempt is made to propagate the ownership from a previous version of a file. This succeeds only if the user has ownership privileges to the previous version.

2. If the attempt to propagate from the previous version fails (either because there is no previous version or the file creator lacks ownership privileges to the previous version), then an attempt is made to propagate ownership from the parent directory. This succeeds only if the user has ownership privileges to the parent directory.
 3. If the attempt to propagate from the parent directory fails, the owner of the file is the same as the creator of the file.
- C39** Ownership of directories is usually set when the directory is created. A user with ownership privileges can specify the owner by using the DCL command `CREATE/DIRECTORY/OWNER_UIC`. If no owner is explicitly defined, then default ownership for a directory is determined as follows:
- 1) If the directory is in alphanumeric or subdirectory format and the user has ownership privileges to that UIC, ownership defaults to the owner UIC of the existing parent directory; if not, the ownership defaults to the UIC of the process issuing the command.
 - 2) If the directory name is in UIC format, ownership defaults to the UIC in the directory name.
- C40** The protection code of a subdirectory defaults to that of the next higher level directory. The default protection code of a top-level directory comes from the volume master file directory.
- C41** A subdirectory inherits the ACL of the parent directory by default.
- C42** The `NOPROPAGATE` option for an ACL can be used to specify which ACEs are not to be copied to newly created subdirectories.
- C43** If a protection code is not defined for a file when it is created, the system applies a default protection code as follows:
1. If the file is a new version of an existing file, the new file inherits the protection of the existing version.
 2. If no previous version exists, and the directory where the file is to be stored has an associated ACL that includes a `DEFAULT_PROTECTION` ACE, the protection specified by that ACE is used.
 3. If the directory does not have a `DEFAULT_PROTECTION` ACE, the default process protection is used. The system-wide default for file protection is specified by the `SYSGEN` parameter `RMS_FILEPROT`. However, the system-wide default can be overridden by using the `SET PROTECTION/DEFAULT` command in a user's login command procedure.
- C44** Default ACL protection for a file is determined as follows:
1. If the file is a new version of an existing file, the new file inherits the ACL of the existing one, without any ACEs marked with the `NOPROPAGATE` option.
 2. If no previous version of the file exists, VMS checks the directory in which the file for the ACL is being created. If there is an ACL, all entries in the ACL marked with the `DEFAULT` option are copied to the new file with the `DEFAULT` option removed.
- C45** The VMS Mail Utility has default file protection to discourage mail tampering.

- C46** The /HANGUP qualifier can be used with the LOGOUT command to automatically break the connection to a dialup line after the logout. The /HANGUP qualifier will work for all terminals that have been set up with the HANGUP terminal characteristic.
- C47** Use of DCL commands can be restricted via the following methods:
1. Remove or copy DCL command definitions, and rebuild the DCL tables. Use the /CLITABLES qualifier in the user's UAF record to specify the modified tables. Specify /FLAGS=DEFCLI to ensure that the user can only log in with the specified CLI and tables, and protect the original DCL tables from unauthorized access.
 2. Impose ACLs on the system program files in the directories SYS\$SYSROOT:[SYSEXE] and SYS\$SYSROOT:[SYSLIB].
- C48** Reuse of system memory pages is protected by the VMS memory management subsystem and cannot be defeated.
- C49** An erasure pattern can be applied as files are deleted or purged. The system manager can apply this method automatically or users can specify the erasure pattern on a file-by-file basis.
- C50** Highwater marking is a technique provided by VMS to keep users from reading file space beyond the areas where they have been permitted to write. The outer limit of written space on the file is that file's highwater mark. This technique keeps users from scavenging portions of the disk for information that they did not actually write. Highwater marking is used on sequential, exclusively accessed files, such as files output from various text editors, compilers, and linkers.
- Highwater marking is enabled by default for a volume when it is initialized, but it can be disabled.
- C51** For indexed and shared sequential files, a technique known as erase-on-allocate is used in place of true highwater marking. Erase-on-allocate means that blocks of disk space are erased as they are allocated to the user.
- C52** VMS performs error checking on register contents to insure instruction integrity. The VMS ERRFMT process records the system errors in the file SYS\$ERRORLOG:ERRLOG.SYS. This file contains hardware device errors, error correction code errors, interrupt exception conditions, machine checks, and software bugchecks. Error correction codes repair single-bit errors and detect double-bit errors.
- C53** The Accounting Utility (ACCOUNTING) uses the data from the system's accounting log file (ACCOUNTING.DAT) to produce reports on system events including process or image termination, system initialization, login failure, and print jobs.
- C54** The system security manager requires the additional SECURITY privilege to enable security auditing and to set up security operator terminals.
- C55** The system security administrator can enable a security operator's terminal to receive messages resulting from security-related events.
- C56** Security auditing can be enabled in the form of security alarms to audit the following events:

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

The operating system writes information resulting from the enabled security alarms to the security operator's log file. The alarm reveals the name of the perpetrator, the method of access, time of access, and the use of a privilege to gain access to the object. The command procedure SECAUDIT.COM is used to selectively extract information from this file.

- C57** Actions taken by trusted users of the system (operators, administrators, and security officers) can be audited by the enforced use of terminal session auditing. Attempts to defeat auditing can be detected by (1) enabling auditing of authorization modifications and (2) placing ACL entries on the captive login command procedures and the directories containing them to detect modification of the procedures.
- C58** VMS provides a file encryption utility for disk files.
- C59** An automatic timeout mechanism has been locally implemented to terminate idle processes.
- C60** Procurement is limited to those companies who are reputable, and bonded, and have signed certification statements declaring their products to be free of subversive code (trojan horses, time bombs, viruses, worm-hole routines, etc.). Where local modifications are required, source code must be furnished or the appropriate hooks provided.
- C61** An ADP Security Plan is submitted to the proper accrediting official and the CSOM for approval before classified information is processed on a computer system. Further, if the system includes a network environment that crosses operations office geographic boundaries, then the plan is approved by the CSPM. This effort is coordinated with the site computer security group.
- C62** A Computer Protection Plan is submitted by the CPPM to the appropriate site management for approval before unclassified sensitive information is processed on a computer system.
- C63** Periodic risk assessment reviews are conducted in accordance with agency orders for unclassified sensitive computers and classified computers.
- C64** Backup procedures are implemented as appropriate to ensure minimal loss of data in the event that data restoration from backup media is necessary. One copy of the data backups is kept in a different location.

- C65 A contingency recovery plan has been developed and is periodically reviewed to ensure that the plan is kept current.
- C66 All vendor-supplied or publicized accounts are disabled or the passwords changed.
- C67 The number of users having access to privileged accounts is held to a minimum. Up-to-date lists of those persons having access to privileged accounts and the accounts themselves are submitted to the appropriate security authority. This list includes users with the ability to grant privileges.
- C68 Initial user passwords are assigned and distributed by an authorized system security official.
- C69 Nonencrypted dial-up access and nonencrypted dial-out capability are not allowed on computers used for processing classified or sensitive information.
- C70 VMS version 4.3 has been certified as a C2 system. Therefore, it is assumed that the documentation requirements for a DOD C2 System have been met.
- C71 VMS version 4.3 has been certified as a C2 system. Therefore, it is assumed that the life cycle requirements for a DOD C2 System have been met.
- C72 A utility called the User Environment Test Package (UETP) is provided to test the performance of system hardware and software. This utility runs test exercises to simulate users running programs that access disk files while checking file system inconsistencies. UETP also performs some minor diagnostics. This utility does not attempt to diagnose the cause of errors; it simply reports detected errors⁴.
- C73 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 the correct function of the microcode and the data path circuitry used in the address translation logic of the virtual memory subsystem⁴.

APPENDIX C
EXPOSURES AND CONTROLS

APPENDIX C: EXPOSURES AND CONTROLS

Controls are listed in Appendix B, "Agency Controls". Exposures were taken from Appendix B, "Agency Requirements and Exposures," and numbered.

- E1 Unauthorized subject allowed access to an object
C23, C24, C25, C26, C27
- E2 Authorized subject denied access to an object
C23, C24, C25, C26, C27, C28, C29
- E3 Propagation of access rights to an object is unlimited, allowing uncontrolled granting of access rights
C37, C42
- E4 Unauthorized subject creates or modifies user profiles
C23, C24, C25, C26, C27, C55, C56
- E5 Authorized subject not allowed to create or modify user profiles
C23, C24, C25, C26, C27, C28, C29
- E6 Unauthorized subject modifies authentication data
C23, C24, C25, C26, C27, C55, C56
- E7 Authorized subject is not allowed to modify authentication data
C23, C24, C25, C26, C27, C28, C29
- E8 Unauthorized subject modifies the operating system
C23, C24, C25, C26, C27, C30, C31, C55, C56
- E9 Authorized subject not allowed to modify the operating system
C23, C24, C25, C26, C27, C28, C29, C30
- E10 Unauthorized subject modifies audit log to cover security violations
C23, C24, C25, C26, C27, C55, C56
- E11 Authorized system security administrator modifies audit log to cover security violations
C55, C56, C57
- E12 Exposure of sensitive or classified information to unauthorized users by dumping areas of memory, including memory assigned to another subject
C31
- E13 Programmer obtains access to data or files of another subject on secondary memory
C23, C24, C25, C26, C27, C33, C34, C35, C49, C50
- E14 Unauthorized subject accesses restricted system resources

- C23, C24, C25, C26, C27, C32, C33, C34, C35
- E15** Authorized subject not allowed access to restricted system resources
C23, C24, C25, C26, C27, C28, C32, C33, C34, C35
- E16** Access control default allows access of objects to subjects without the need to know
C38, C39, C40, C41, C42, C43, C44, C45
- E17** Nonuser allowed access to system information
C8, C18, C19
- E18** Subject space in main memory is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject
C48
- E19** Subject space in secondary storage is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject
C49, C50
- E20** System security administrator is unaware of unauthorized creation or modification of user profiles
C55, C56
- E21** Sensitive or classified information is channeled to an unauthorized subject
- E22** Nonuser allowed access to the system
C1, C2, C3, C4, C5, C6, C7, C8, C10, C11, C13, C17, C59, C66
- E23** Repeated unsuccessful logon attempts are not noticed by the system security administrator
C53, C55, C56
- E24** System can not authenticate user identity
C8, C10, C11
- E25** The system cannot associate the user with a user profile
C8, C25
- E26** Passwords are displayed to unauthorized or nonusers
C9, C12
- E27** Plain text authentication data is stored in the computer system and exposed to unauthorized or nonuser
C12
- E28** Encrypted passwords are exposed to an unauthorized user who is able to decrypt them
C12
- E29** Users with the ability to choose their own passwords choose simplistic passwords that are easily guessed by an unauthorized or nonuser
C14, C15, C16

- E30** User accesses the system from an unauthorized device
C25, C26, C35
- E31** User not allowed to access the system from an authorized device
C25, C26, C28, C35
- E32** User is unaware of unauthorized access to his/her user account
C20
- E33** Exported object is exported without the appropriate sensitivity label
- E34** Hardcopy output is produced without readable labels identifying the sensitivity of the information
- E35** Data is entered into the system with no, or an incorrect, associated sensitivity label
- E36** Unauthorized subject modifies the classification level or sensitivity label of an object
- E37** Authorized subject not allowed to modify the classification level or sensitivity label of an object
- E38** The system cannot associate an individual user with audited events
C25, C56
- E39** The system fails to record sufficient information to detect attempted security violations
C26, C56
- E40** Unauthorized access to an object is not logged
C26, C53, C56, C57
- E41** The system fails to audit the minimum set of identified areas of potential security violations
C26, C53, C55, C56, C57
- E42** Necessary auditing capability not provided
C26, C53, C55, C56, C57
- E43** User mode program accesses main memory space of another user or the system
C30, C31
- E44** User mode program accesses secondary storage space of another user or the system
C23, C24, C25, C26, C27, C30
- E45** User mode program accesses privileged functions
C30
- E46** Unauthorized subject executes privileged instructions
C23, C24, C25, C26, C27, C30, C47

- E47 Authorized subject not allowed to execute privileged instructions
C23, C24, C25, C26, C27, C30, C47
- E48 User mode program acquires access to input/output device, memory management functions, user process executing characteristics
C30
- E49 Execution of a command produces unpredictable, potentially detrimental, results
C52, C72, C73
- E50 Programmer inserts program that masquerades as security program and obtains user ID and password of another user
C22
- E51 Nonuser allowed unlimited trail-and-error attempts to logon to the system
C13, C17
- E52 Automatic timeout of process is disabled
C23, C24, C25, C26, C27
- E53 User terminal cannot be locked out by the system administrator
C26, C35
- E54 Power-up of terminal preserves pre-existing user session
- E55 Dial-up access is not monitored, resulting in unknown external access to the system
C1, C26, C55, C56, C69
- E56 Dial-up terminal becomes disconnected, leaving the session open for any user who connects to that port
C46, C59
- E57 Unauthorized user gains access to a privileged account and assumes system control
C5, C17, C66
- E58 Sensitivity labels associated with classified objects are corrupted, allowing access by unauthorized subjects or access denial to authorized subjects
- E59 User leaves logged-on terminal unattended for extended periods of time
C59
- E60 Non-obvious security exposures exist in the system
C71
- E61 Security exposures exist due to lack of comprehensive consideration of all possible exposures
C71

- E62** Misuse or nonuse of security features due to lack of functional understanding
C71
- E63** Uncontrolled modification result in inconsistency among components of the system
C71
- E64** Introduction of security exposures through object modification
C71
- E65** Security exposures exist due to the lack of understanding of the interface between system modules
C71
- E66** System software has not been sufficiently tested by the proper agencies to ensure that security exposures do not exist.
C71
- E67** Computer system used for processing classified data has not been approved for processing of classified data
C61
- E68** Computer system used for processing unclassified sensitive data has not been approved for processing of unclassified sensitive data
C62
- E69** Changes to the system or the physical environment produce new exposures that have not been considered in the current security plan
C63
- E70** Data is lost due to the restoration of backup data
C64
- E71** Data backups are not properly maintained, resulting in excessive data loss in the event of media damage
C64
- E72** Some event (i.e., natural disaster) yields the system inoperable and there is no means to continue processing
C65
- E73** User access to privileged accounts is not monitored, thus users have access to privileged accounts without an established need
C67

APPENDIX D
TRANSACTION TEST SCRIPTS

APPENDIX D: TRANSACTION TEST SCRIPTS

READ Transaction

Steps 1 through 8 (Logon)

See Logon Subsystem transaction flow diagram.

Step 9 (User Process)

E18 Subject space in main memory is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject

C48

E19 Subject space in secondary storage is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject

C49, C50

E52 Automatic timeout of process is disabled

C23, C24, C25, C26, C27

E54 Power-up of terminal preserves pre-existing user session

E56 Dial-up terminal becomes disconnected, leaving the session open for any user who connects to that port

C46, C59

E59 User leaves logged-on terminal unattended for extended periods of time

C59

E67 Computer system used for processing classified data has not been approved for processing of classified data

C61

E68 Computer system used for processing unclassified sensitive data has not been approved for processing of unclassified sensitive data

C62

Steps 10 through 13 (Access Control)

See Access Control Subsystem transaction flow diagram.

WRITE Transaction**Steps 1 through 8 (Logon)**

See Logon Subsystem transaction flow diagram.

Step 9 (User Process)

- E18** Subject space in main memory is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject

C48

- E19** Subject space in secondary storage is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject

C49, C50

- E52** Automatic timeout of process is disabled

C23, C24, C25, C26, C27

- E54** Power-up of terminal preserves pre-existing user session

- E56** Dial-up terminal becomes disconnected, leaving the session open for any user who connects to that port

C46, C59

- E59** User leaves logged-on terminal unattended for extended periods of time

C59

- E67** Computer system used for processing classified data has not been approved for processing of classified data

C61

- E68** Computer system used for processing unclassified sensitive data has not been approved for processing of unclassified sensitive data

C62

Steps 10 through 13 (Access Control)

See Access Control Subsystem transaction flow diagram.

EXECUTE Transaction**Steps 1 through 8 (Logon)**

See Logon Subsystem transaction flow diagram.

Step 9 (User Process)

- E18** Subject space in main memory is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject

C48

E19 Subject space in secondary storage is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject

C49, C50

E52 Automatic timeout of process is disabled

C23, C24, C25, C26, C27

E54 Power-up of terminal preserves pre-existing user session

E56 Dial-up terminal becomes disconnected, leaving the session open for any user who connects to that port

C46, C59

E59 User leaves logged-on terminal unattended for extended periods of time

C59

E67 Computer system used for processing classified data has not been approved for processing of classified data

C61

E68 Computer system used for processing unclassified sensitive data has not been approved for processing of unclassified sensitive data

C62

Steps 10 through 13 (Access Control)

See Access Control Subsystem transaction flow diagram.

DELETE Transaction

Steps 1 through 8 (Logon)

See Logon Subsystem transaction flow diagram.

Step 9 (User Process)

E18 Subject space in main memory is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject

C48

E19 Subject space in secondary storage is not sanitized prior to reassignment to another subject, resulting in access by an unauthorized subject

C49, C50

E52 Automatic timeout of process is disabled

C23, C24, C25, C26, C27

E54 Power-up of terminal preserves pre-existing user session

E56 Dial-up terminal becomes disconnected, leaving the session open for any user who connects to that port

C46, C59

E59 User leaves logged-on terminal unattended for extended periods of time

C59

E67 Computer system used for processing classified data has not been approved for processing of classified data

C61

E68 Computer system used for processing unclassified sensitive data has not been approved for processing of unclassified sensitive data

C62

Steps 10 through 13 (Access Control)

See Access Control Subsystem transaction flow diagram.

Logon Subsystem

Step 1 (Connect to system)

E53 User terminal cannot be locked out by the system administrator

C26, C35

E54 Power-up of terminal preserves pre-existing user session

E55 Dial-up access is not monitored, resulting in unknown external access to the system

C1, C26, C55, C56, C69

E56 Dial-up terminal becomes disconnected, leaving the session open for any user who connects to that port

C46, C59

Step 2 (Enter system password)

E22 Nonuser allowed access to the system

C1, C2, C3, C4, C5, C6, C7, C8, C10, C11, C13, C17, C59, C66

E30 User accesses the system from an unauthorized device

C25, C26, C35

E31 User not allowed to access the system from an authorized device

C25, C26, C28, C35

Step 3 (Display announcement message)

E17 Nonuser allowed access to system information

C8, C18, C19

Step 4 (Enter user name)**E22** Nonuser allowed access to the system

C1, C2, C3, C4, C5, C6, C7, C8, C10, C11, C13, C17, C59, C66

E23 Repeated unsuccessful logon attempts are not noticed by the system security administrator

C53, C55, C56

E25 The system cannot associate the user with a user profile

C8, C25

E30 User accesses the system from an unauthorized device

C25, C26, C35

E31 User not allowed to access the system from an authorized device

C25, C26, C28, C35

E38 The system cannot associate an individual user with audited events

C25, C56

E50 Programmer inserts program that masquerades as security program and obtains user ID and password of another user

C22

E51 Nonuser allowed unlimited trail-and-error attempts to logon to the system

C13, C17

E57 Unauthorized user gains access to a privileged account and assumes system control

C5, C17, C66

E73 User access to privileged accounts is not monitored, thus users have access to privileged accounts without an established need

C67

Step 5 (Enter primary password)**E22** Nonuser allowed access to the system

C1, C2, C3, C4, C5, C6, C7, C8, C10, C11, C13, C17, C59, C66

E24 System can not authenticate user identity

C8, C10, C11

E26 Passwords are displayed to unauthorized or nonusers

C9, C12

E27 Plaintext authentication data is stored in the computer system and exposed to unauthorized or nonuser

C12

E28 Encrypted passwords are exposed to an unauthorized user who is able to decrypt them

C12

E29 Users with the ability to choose their own passwords choose simplistic passwords that are easily guessed by an unauthorized or nonuser

C14, C15, C16

E57 Unauthorized user gains access to a privileged account and assumes system control

C5, C17, C66

E73 User access to privileged accounts is not monitored, thus users have access to privileged accounts without an established need

C67

Step 6 (Enter secondary password)

E22 Nonuser allowed access to the system

C1, C2, C3, C4, C5, C6, C7, C8, C10, C11, C13, C17, C59, C66

E24 System can not authenticate user identity

C8, C10, C11

E26 Passwords are displayed to unauthorized or nonusers

C9, C12

E27 Plaintext authentication data is stored in the computer system and exposed to unauthorized or nonuser

C12

E28 Encrypted passwords are exposed to an unauthorized user who is able to decrypt them

C12

E29 Users with the ability to choose their own passwords choose simplistic passwords that are easily guessed by an unauthorized or nonuser

C14, C15, C16

E57 Unauthorized user gains access to a privileged account and assumes system control

C5, C17, C66

E73 User access to privileged accounts is not monitored, thus users have access to privileged accounts without an established need

C67

Step 7 (Display welcome message)

E17 Nonuser allowed access to system information
C8, C18, C19

Step 8 (Display last logon message)

E32 User is unaware of unauthorized access to his/her user account
C20

Access Control Subsystem**Step 1 (Perform access mode check)**

E1 Unauthorized subject allowed access to an object
C23, C24, C25, C26, C27

E2 Authorized subject denied access to an object
C23, C24, C25, C26, C27, C28, C29

E8 Unauthorized subject modifies the operating system
C23, C24, C25, C26, C27, C30, C31, C55, C56

E9 Authorized subject not allowed to modify the operating system
C23, C24, C25, C26, C27, C28, C29, C30

E12 Exposure of sensitive or classified information to unauthorized users by dumping areas of memory, including memory assigned to another subject
C31

E14 Unauthorized subject accesses restricted system resources
C23, C24, C25, C26, C27, C32, C33, C34, C35

E15 Authorized subject not allowed access to restricted system resources
C23, C24, C25, C26, C27, C28, C32, C33, C34, C35

E43 User mode program accesses main memory space of another user or the system
C30

E44 User mode program accesses secondary storage space of another user or the system
C23, C24, C25, C26, C27, C30

E45 User mode program accesses privileged functions
C30

E46 Unauthorized subject executes privileged instructions
C23, C24, C25, C26, C27, C30, C47

E47 Authorized subject not allowed to execute privileged instructions

C23, C24, C25, C26, C27, C30, C47

E48 User mode program acquires access to input/output device, memory management functions, user process executing characteristics

C30

Step 2 (If the object has an ACL, then check ACL protection.)

E1 Unauthorized subject allowed access to an object

C23, C24, C25, C26, C27

E2 Authorized subject denied access to an object

C23, C24, C25, C26, C27, C28, C29

E3 Propagation of access rights to an object is unlimited, allowing uncontrolled granting of access rights

C37, C42

E4 Unauthorized subject creates or modifies user profiles

C23, C24, C25, C26, C27, C55, C56

E5 Authorized subject not allowed to create or modify user profiles

C23, C24, C25, C26, C27, C28, C29

E6 Unauthorized subject modifies authentication data

C23, C24, C25, C26, C27, C55, C56

E7 Authorized subject is not allowed to modify authentication data

C23, C24, C25, C26, C27, C28, C29

E8 Unauthorized subject modifies the operating system

C23, C24, C25, C26, C27, C30, C31, C55, C56

E9 Authorized subject not allowed to modify the operating system

C23, C24, C25, C26, C27, C28, C29, C30

E10 Unauthorized subject modifies audit log to cover security violations

C23, C24, C25, C26, C27, C55, C56

E11 Authorized system security administrator modifies audit log to cover security violations

C55, C56, C57

E13 Programmer obtains access to data or files of another subject on secondary memory

C23, C24, C25, C26, C27, C33, C34, C35, C49, C50

E14 Unauthorized subject accesses restricted system resources

C23, C24, C25, C26, C27, C32, C33, C34, C35

E15 Authorized subject not allowed access to restricted system resources

C23, C24, C25, C26, C27, C28, C32, C33, C34, C35

E16 Access control default allows access of objects to subjects without the need to know

C38, C39, C40, C41, C42, C43, C44, C45

E30 User accesses the system from an unauthorized device

C25, C26, C35

E31 User not allowed to access the system from an authorized device

C25, C26, C28, C35

E39 The system fails to record sufficient information to detect attempted security violations

C26, C56

E40 Unauthorized access to an object is not logged

C26, C53, C56, C57

E41 The system fails to audit the minimum set of identified areas of potential security violations

C26, C53, C55, C56, C57

E42 Necessary auditing capability not provided

C26, C53, C55, C56, C57

Step 3 (If the requested access is not explicitly denied in an ACL or if the subject UIC matches the OWNER or SYSTEM fields of the object UIC, then check UIC-based protection)

E1 Unauthorized subject allowed access to an object

C23, C24, C25, C26, C27

E2 Authorized subject denied access to an object

C23, C24, C25, C26, C27, C28, C29

E3 Propagation of access rights to an object is unlimited, allowing uncontrolled granting of access rights

C37, C42

E4 Unauthorized subject creates or modifies user profiles

C23, C24, C25, C26, C27, C55, C56

E5 Authorized subject not allowed to create or modify user profiles

C23, C24, C25, C26, C27, C28, C29

E6 Unauthorized subject modifies authentication data

C23, C24, C25, C26, C27, C55, C56

E7 Authorized subject is not allowed to modify authentication data

C23, C24, C25, C26, C27, C28, C29

E8 Unauthorized subject modifies the operating system

C23, C24, C25, C26, C27, C30, C31, C55, C56

E9 Authorized subject not allowed to modify the operating system

C23, C24, C25, C26, C27, C28, C29, C30

E10 Unauthorized subject modifies audit log to cover security violations

C23, C24, C25, C26, C27, C55, C56

E12 Exposure of sensitive or classified information to unauthorized users by dumping areas of memory, including memory assigned to another subject

C31

E13 Programmer obtains access to data or files of another subject on secondary memory

C23, C24, C25, C26, C27, C33, C34, C35, C49, C50

E14 Unauthorized subject accesses restricted system resources

C23, C24, C25, C26, C27, C32, C33, C34, C35

E15 Authorized subject not allowed access to restricted system resources

C23, C24, C25, C26, C27, C28, C32, C33, C34, C35

E16 Access control default allows access of objects to subjects without the need to know

C38, C39, C40, C41, C42, C43, C44, C45

E30 User accesses the system from an unauthorized device

C25, C26, C35

E31 User not allowed to access the system from an authorized device

C25, C26, C28, C35

Step 4 (Check subject for privileges.)

E1 Unauthorized subject allowed access to an object

C23, C24, C25, C26, C27

E2 Authorized subject denied access to an object

C23, C24, C25, C26, C27, C28, C29

E4 Unauthorized subject creates or modifies user profiles

C23, C24, C25, C26, C27, C55, C56

E5 Authorized subject not allowed to create or modify user profiles

C23, C24, C25, C26, C27, C28, C29

E6 Unauthorized subject modifies authentication data

C23, C24, C25, C26, C27, C55, C56

E7 Authorized subject is not allowed to modify authentication data

C23, C24, C25, C26, C27, C28, C29

- E8** Unauthorized subject modifies the operating system
C23, C24, C25, C26, C27, C30, C31, C55, C56
- E9** Authorized subject not allowed to modify the operating system
C23, C24, C25, C26, C27, C28, C29, C30
- E10** Unauthorized subject modifies audit log to cover security violations
C23, C24, C25, C26, C27, C55, C56
- E12** Exposure of sensitive or classified information to unauthorized users by dumping areas of memory, including memory assigned to another subject
C31
- E13** Programmer obtains access to data or files of another subject on secondary memory
C23, C24, C25, C26, C27, C33, C34, C35, C49, C50
- E14** Unauthorized subject accesses restricted system resources
C23, C24, C25, C26, C27, C32, C33, C34, C35
- E15** Authorized subject not allowed access to restricted system resources
C23, C24, C25, C26, C27, C28, C32, C33, C34, C35
- E43** User mode program accesses main memory space of another user or the system
C30
- E44** User mode program accesses secondary storage space of another user or the system
C23, C24, C25, C26, C27, C30
- E45** User mode program accesses privileged functions
C30
- E46** Unauthorized subject executes privileged instructions
C23, C24, C25, C26, C27, C30, C47
- E47** Authorized subject not allowed to execute privileged instructions
C23, C24, C25, C26, C27, C30, C47
- E48** User mode program acquires access to input/output device, memory management functions, user process executing characteristics
C30

VITA

A native of Harriman, Tennessee, Susan E. Hicks received a Bachelor of Arts degree in Computer Science from the University of Tennessee, Knoxville in March 1982. She attended graduate school on a part time basis to earn her Master of Science degree in Computer Science while employed in the government contracting industry. She is currently an employee of the Data Systems Research and Development Division of Martin Marietta Energy Systems in Oak Ridge, Tennessee where she completed her thesis work.