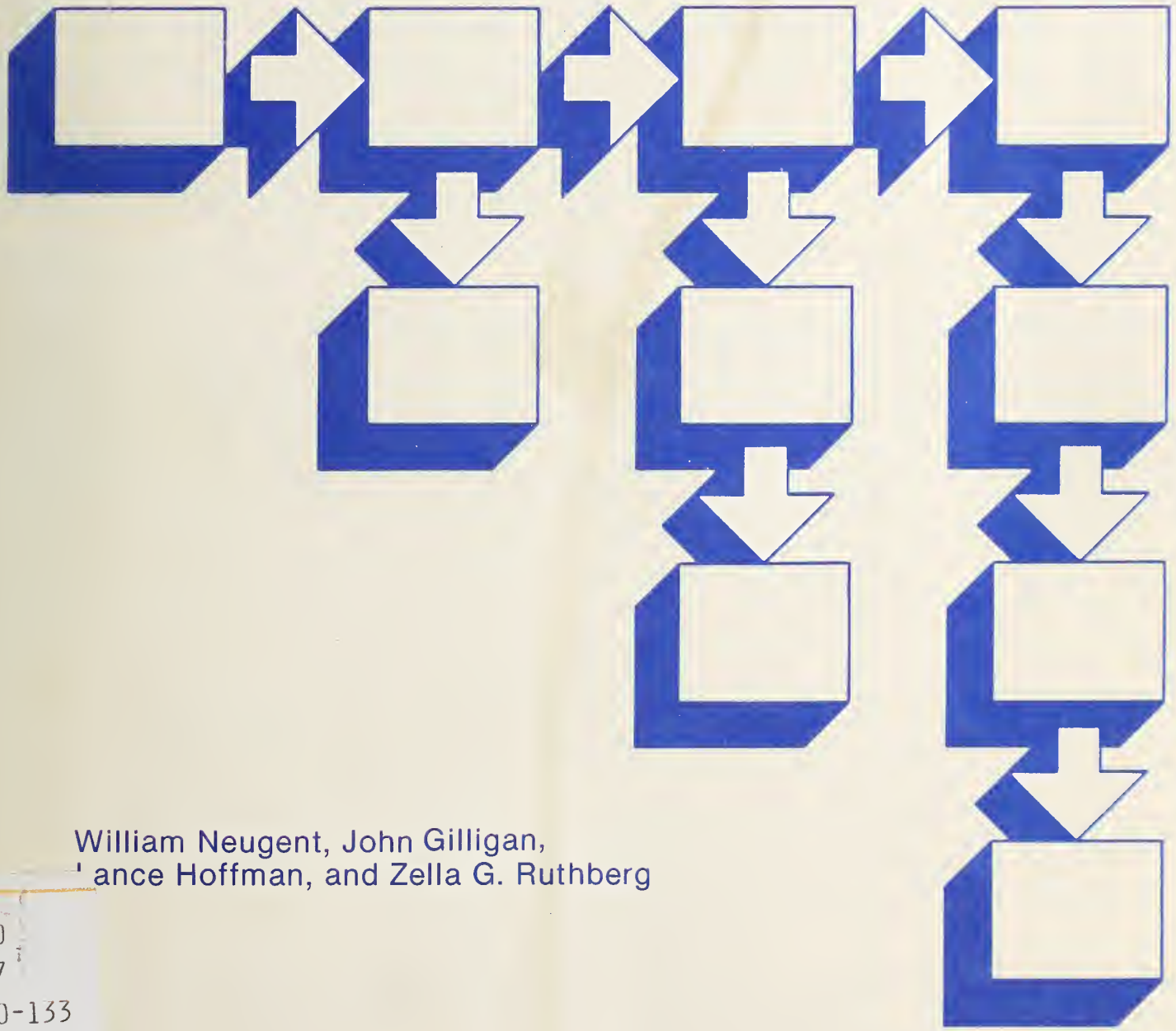
Computer Science and Technology

National Bureau of Standards

NBS Special Publication 500-133

Technology Assessment: Methods for Measuring the Level of Computer Security



William Neugent, John Gilligan,
Lance Hoffman, and Zella G. Ruthberg

The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, the Institute for Computer Sciences and Technology, and the Institute for Materials Science and Engineering.

The National Measurement Laboratory

Provides the national system of physical and chemical measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; provides advisory and research services to other Government agencies; conducts physical and chemical research; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

- Basic Standards[2]
- Radiation Research
- Chemical Physics
- Analytical Chemistry

The National Engineering Laboratory

Provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

- Applied Mathematics
- Electronics and Electrical Engineering[2]
- Manufacturing Engineering
- Building Technology
- Fire Research
- Chemical Engineering[2]

The Institute for Computer Sciences and Technology

Conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

- Programming Science and Technology
- Computer Systems Engineering

The Institute for Materials Science and Engineering

Conducts research and provides measurements, data, standards, reference materials, quantitative understanding and other technical information fundamental to the processing, structure, properties and performance of materials; addresses the scientific basis for new advanced materials technologies; plans research around cross-country scientific themes such as nondestructive evaluation and phase diagram development; oversees Bureau-wide technical programs in nuclear reactor radiation research and nondestructive evaluation; and broadly disseminates generic technical information resulting from its programs. The Institute consists of the following Divisions:

- Inorganic Materials
- Fracture and Deformation[3]
- Polymers
- Metallurgy
- Reactor Radiation

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Gaithersburg, MD 20899.
[2]Some divisions within the center are located at Boulder, CO 80303.
[3]Located at Boulder, CO, with some elements at Gaithersburg, MD.

Computer Science and Technology

NBS Special Publication 500-133

Technology Assessment: Methods for Measuring the Level of Computer Security

William Neugent and John Gilligan

System Development Corporation
McLean, VA 22101

Lance Hoffman

George Washington University
Washington, DC 20052

Zella G. Ruthberg

Center for Programming Science and Technology
Institute for Computer Sciences and Technology
National Bureau of Standards
Gaithersburg, MD 20899

Issued October 1985

U.S. DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Bureau of Standards
Ernest Ambler, Director

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

Library of Congress Catalog Card Number: 85-600600
National Bureau of Standards Special Publication 500-133
Natl. Bur. Stand. (U.S.), Spec. Publ. 500-133, 216 pages (Oct. 1985)
CODEN: XNBSAV

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1985

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROGRAM BACKGROUND 1-1
1.1.1 NBS Computer Security, Integrity, And Risk Management Standards Program 1-1
1.1.2 Security Measurement, Certification, And Accreditation 1-3
1.2 BASIC TERMINOLOGY 1-2
1.3 TECHNOLOGY ASSESSMENT OVERVIEW 1-6
1.3.1 Scope 1-6
1.3.2 Organization 1-6
1.3.3 Sources And Approach 1-7

CHAPTER 2 ENVIRONMENTS

2.1 ENVIRONMENTAL FACTORS INFLUENCING SYSTEM SECURITY 2-1
2.1.1 An Early Approach (1977) 2-1
2.1.2 A More Detailed Later Approach (1979) 2-3
2.1.3 Conclusions About Environmental Factors And System Security 2-7
2.2 ENVIRONMENTAL FACTORS INFLUENCING SECURITY EVALUATION 2-7
2.3 CONCLUSIONS 2-9

CHAPTER 3 CONTROLS

3.1 CONTROL GROUPINGS OR STRUCTURES 3-1
3.1.1 Systems Auditability And Control (SAC) Study [IIA77-2] 3-2
3.1.2 Security: Checklist For Computer Center Self-Audits [AFI79] 3-7
3.1.3 Army Regulation 380-380 [USA79] 3-9
3.1.4 Computer Control Guidelines [CIC70] 3-12
3.1.5 Information Security Handbook [WIL80] 3-12
3.1.6 Control Objectives 1980 [EAF80] 3-13
3.1.7 GAO Internal Controls Course [GAO80] 3-14
3.1.8 NBS SP 500-57 [RUT80] 3-14
3.1.9 EDP Insurance [MIG80] 3-16
3.1.9.1 EDP All Risk 3-16
3.1.9.2 EDP Fidelity 3-16
3.1.9.3 Criteria For Insurance Selection 3-17
3.2 SUMMARY OF CONTROL GROUPINGS OR STRUCTURES . . . 3-17
3.2.1 Roles Served By Control Groupings Or Structures 3-17
3.2.2 General Control Structure 3-20
3.3 EXPOSURE GROUPINGS OR STRUCTURES 3-22

CHAPTER 4 SYSTEM EVALUATION METHODOLOGIES

4.1 SECURITY EVALUATION METHODOLOGIES 4-1
4.1.1 Touche Ross & Co. [MAI76] 4-2
4.1.1.1 Description 4-2

4.1.1.2 Distinguishing Features 4-13
4.1.1.3 Notable Experiences And Lessons 4-13
4.1.1.4 Major Strengths And Weaknesses 4-13
4.1.2 Peat Marwick Mitchell & Co. [PMM80] 4-14
4.1.2.1 Description 4-14
4.1.2.2 Distinguishing Features 4-16
4.1.2.3 Notable Experiences And Lessons 4-16
4.1.2.4 Major Strengths And Weaknesses 4-16
4.1.3 SRI International/USC Info. Sciences Institute (ISI) [NEM78] 4-17
4.1.3.1 Description 4-17
4.1.3.2 Distinguishing Features 4-22
4.1.3.3 Notable Experiences And Lessons 4-22
4.1.3.4 Major Strengths And Weaknesses 4-22
4.1.4 Department Of Defense (DoD) [DOD83] 4-22
4.1.4.1 Description 4-22
4.1.4.2 Distinguishing Features 4-33
4.1.4.3 Notable Experiences And Lessons 4-33
4.1.4.4 Major Strengths And Weaknesses 4-33
4.1.5 Testing 4-34
4.1.5.1 External Testing 4-35
4.1.5.1.1 Independent Definition 4-36
4.1.5.1.2 Pass/Fail Criteria 4-36
4.1.5.2 Internal Testing 4-37
4.1.5.2.1 Measures Of Coverage 4-39
4.1.5.2.2 Software Quality Metrics 4-40
4.1.5.3 Testing For Security Evaluation 4-41
4.1.6 Other Approaches 4-43
4.1.6.1 Canadian Institute Of Chartered Accountants [CIC75] 4-43
4.1.6.2 Arthur Andersen & Co. [AAC78] 4-48
4.1.6.3 AFIPS Security Checklist For Computer Center Self-Audits [AFI79] 4-52
4.1.6.4 Internal Controls For Computerized System [FIT78] 4-55
4.1.6.5 Coopers & Lybrand [C&L82] [HAL85] 4-56
4.1.6.6 Auditing Computer Systems [PER81] 4-58
4.1.6.7 Information Security Handbook [WIL80] 4-59
4.1.6.8 Department Of Health And Human Services [HHS78] [HHS82] 4-60
4.1.6.9 Department Of Agriculture [DOA80] [DOA84] . . 4-61
4.1.6.10 GAO Audit Guides [GAO81-1] [GAO81-2] 4-63
4.1.6.11 Department Of Energy [DOE83] [DOE84] 4-65
4.1.6.12 Formal Verification 4-68
4.2 RISK ASSESSMENT METHODOLOGIES 4-72
4.2.1 Individual Methodologies 4-72
4.2.1.1 FIPS PUB 65 [FIP65] 4-73
4.2.1.2 Air Force Risk Analysis Management Program (AFRAMP) [AFRAMP] 4-74
4.2.1.3 Department Of Agriculture [DOA77] 4-75
4.2.1.4 SDC Navy Risk Assessment Methodology [SDC79] . 4-75
4.2.1.5 Risk Analysis And Management Program (IST/RAMP) [[IST79] 4-76
4.2.1.6 Relative Impact Measure (RIM) Of Vulnerability [NIE80] 4-81

4.2.1.7 Fuzzy Risk Analysis [HOF80] 4-82
4.2.1.8 Security Assessment Questionnaire [IBM80] [IBM85] 4-85
4.2.2 Evaluation Of Risk Assessment Methodologies . 4-85
4.2.3 Generic Problems With Existing Risk Assessment Methodologies 4-87

CHAPTER 5 SUMMARY OF THE STATE OF THE ART

5.1 SIMILARITIES AND DIFFERENCES 5-1
5.2 DIFFERENCES DERIVING FROM METHODOLOGICAL PURPOSE . 5-3
5.2.1 Basic Purposes And Elements 5-3
5.2.2 Analytic Emphasis 5-5
5.2.3 Quantification 5-8
5.3 DIFFERENCES DERIVING FROM EVALUATION OBJECTIVE . . 5-8
5.3.1 Flaws Versus Flaw Susceptibility 5-9
5.3.2 Anticipated Versus Unanticipated Threats And Attacks 5-10
5.3.3 Operation Versus Development 5-11
5.4 COMMON APPROACHES FOR STRUCTURING ANALYSIS . . . 5-12
5.4.1 Matrix 5-13
5.4.2 Checklist 5-15
5.4.3 Transaction Flow 5-17
5.4.4 Loosely Structured 5-18
5.4.5 Focusing And Other Issues 5-18
5.5 GENERAL EVALUATION ISSUES 5-19
5.5.1 Quantification 5-20
5.5.2 Uncertainty And Bias 5-24
5.5.3 Integration With The Decision Process 5-28
5.6 CONCLUSIONS 5-29
5.6.1 On Measuring Levels Of Computer Security . . . 5-30
5.6.2 Need For Guidance On Security Evaluation . . . 5-30
5.6.3 On Selecting A Methodology 5-30
5.6.4 On Selecting The Purpose Of The Methodology . 5-31
5.6.5 On Tailoring The Depth Of The Evaluation . . . 5-31
5.6.6 On The Importance Of Skilled Evaluators . . . 5-31

CHAPTER 6 SECURITY POLICY IMPACT

6.1 SENSITIVITY DISTINCTIONS 6-1
6.2 ACCEPTANCE CRITERIA 6-4
6.2.1 Definition Of Acceptance Criteria 6-5
6.2.2 Security Requirement Classes For Evaluation Analysis 6-10
6.2.3 Types Of Acceptance Criteria 6-13
6.2.4 Conclusions 6-20

CHAPTER 7 DOCUMENT OVERVIEW

7.1 INTRODUCTION 7-1
7.2 ENVIRONMENTS 7-2
7.3 CONTROLS 7-2
7.3.1 Control Groupings Or Structures 7-2
7.3.2 Exposure Groupings Or Structures 7-3
7.4 EVALUATION METHODOLOGIES/APPROACHES 7-3

7.5 SUMMARY OF THE STATE-OF-THE-ART 7-8
7.5.1 Similarities And Differences 7-8
7.5.2 Approaches For Structuring Analysis 7-9
7.5.3 General Evaluation Issues 7-9
7.5.4 Conclusions 7-10
7.6 SECURITY POLICY IMPACT 7-10
7.6.1 Sensitivity Distinctions 7-10
7.6.2 Acceptance Criteria 7-11
7.6.3 Conclusions 7-11

APPENDIX A REFERENCES A-1

LIST OF TABLES

Table		Page
2-1	Environmental Parameter Categories	2-2
2-2	Dedicated Data Base Management System (e.g., Airline Reservations)	2-4
2-3	Environmental Influences	2-6
3-1	Applications Systems Controls from the Systems Auditability and Control Study	3-3
3-2	Computer Service Center Control Structure	3-5
3-3	Application System Development Control Structure	3-6
3-4	AFIPS Guideline Evaluation Categories	3-8
3-5	Control Evaluation Guide of the Canadian Institute of Chartered Accountants	3-11
3-6	Illustrative General Control Structure	3-21
3-7	Mutually Exclusive Exposures	3-23
3-8	Overlapping Exposures	3-24
4-1	Touche Ross Application Activities Subject to Control	4-5
4-2	Exposures	4-6
4-3	Relationships of Activities Subject to Control to Causes of Exposure	4-8
4-4	Application Relationships of Causes and Exposures	4-9
4-5	Application Control Evaluation Table	4-10
4-6	Categories of Protection Flaws (and examples) (Based on Bisbey, Carlstedt, Hollingworth at ISI)	4-19
4-7	Symptoms of Potential Protection Flaws, by Category	4-19
4-8	Factors Influencing Defensiveness in Systems and Applications	4-20
4-9	Evaluation of Multics and UNIX with Respect to Characteristic Flaws	4-21
4-10	Evaluation of Multics and UNIX with Respect to Methodological Considerations	4-21
4-11	Secure System Categories	4-28
4-12	Protection Levels	4-30
4-13	Loss Category, Asset Type, Interaction Function Relation	4-78
5-1	Differences Deriving from Purpose	5-4
5-2	Comparison of Common Approaches for Structuring Analysis	5-14

LIST OF FIGURES

Figure		Page
3-1	A Model for Categorizing System Vulnerabilities and Managerial Controls According to Responsible Organizational Units	3-15
4-1	PRE 1981 Approval for DoD Use	4-25
4-2	POST 1981 Approval for DoD Use	4-25
4-3	Security Metric	4-27
4-4	Attributes of Trusted Operating System	4-29
4-5	Protection-Level User Chart with Added Limitations	4-32
4-6	Software Quality Factors	4-42
4-7	Control Evaluation Guide Summary	4-45
4-8	Control Evaluation Guide Excerpt	4-46
4-9	Verification Guide Excerpt	4-47
4-10	Control Deficiency Guide Excerpt	4-49
4-11	Illustrative RAMP Reports	4-79
4-12	Typical RAMP Report	4-80
4-13	Input Data to Fuzzy Risk Analysis for a Hypothetical Computer Center	4-83
4-14	Example of Fuzzy Risk Analysis Output	4-84
6-1	Illustrative Definition of Acceptable Rates of Loss or Degradation	6-8
6-2	Requirement/Evaluation Classes and Objectives	6-11
6-3	Sample Penetration Resistance Acceptability Criteria	6-19

PREFACE

William Neugent is the principal author of the original draft report submitted, under contract, by System Development Corporation (SDC). Mr. Neugent's work was performed under the supervision of John Gilligan at SDC. Dr. Lance Hoffman of George Washington University wrote Section 4.2 on Risk Assessment Methodologies for the original draft report, under contract to SDC. Zella Ruthberg of the National Bureau of Standards (NBS) developed and directed the NBS project on Computer Security Certification and Accreditation within which this SDC effort took place. Ms. Ruthberg undertook the updating and editing of the draft report. This resulted in adding descriptions of several approaches and methodologies that have appeared in the interim, as well as changing wording and formats where considered appropriate.

It is the intent that this document will be a useful companion to FIPS PUB 102, Guideline for Computer Security Certification and Accreditation. Certification of the security status of a computer application, system, or installation depends on a technical security evaluation. This document provides pointers to and descriptions of a large number of the approaches and methodologies in use today. Although risk assessment methodologies on microcomputers were considered to be beyond the scope of this document, pp. 4-72,73 contain pointers to current methodologies in that arena.

CHAPTER 1

INTRODUCTION

This technology assessment constitutes a summary and assessment of methods for measuring the level of computer security in computer applications, systems, and installations. The initial draft report for this document was produced in June 1981 for the National Bureau of Standards (NBS) by the System Development Corporation (SDC) as part of the NBS Computer Security and Risk Management Standards Program. The intent of that report was to provide a comprehensive assessment of the state of the art and to provide a suitable basis for producing a Federal Information Processing Standards Publication (FIPS PUB) guideline on computer security, certification, and accreditation. The FIPS PUB guideline was subsequently developed and issued as FIPS PUB 102 on September 27, 1983 and titled "Guidelines for Computer Security Certification and Accreditation" [FIP102]. This technology assessment is now being issued as a companion foundation document to FIPS PUB 102. The initial draft report has been brought up to date by changing some methodology discussions, adding a few methodologies, referencing relevant documents that appeared in the interim, and modifying the text where appropriate.

1.1 PROGRAM BACKGROUND

For the reader who may be unfamiliar with NBS' computer security, integrity, and risk management program, a brief overview is provided here. The need for a technology assessment report dealing with methods for measuring computer security and some motivation for the guideline on computer security certification and accreditation are also discussed.

1.1.1 NBS Computer Security, Integrity, And Risk Management Standards Program

The National Bureau of Standards (NBS) through its Institute for Computer Sciences and Technology (ICST) initiated a Computer Security and Risk Management program in 1972. Since that time, numerous standards, guidelines, and technical reports have been issued in the

areas of physical security, technical security, and computer security management.

ICST has structured a comprehensive program in computer integrity (detecting unauthorized entry or change of information), confidentiality (preventing unauthorized disclosure of information), and reliability (assuring availability of information processing) to reduce existing vulnerabilities and risks. The program encompasses research and development of security standards, transfer of technology to potential implementors and vendors, and assistance to users of security technology.

ICST draws upon its own research and that of other organizations in accomplishing its goals. Technology transfer interfaces have been established linking vendors and users, government and industry, managers and technologists.

1.1.2 Security Measurement, Certification, And Accreditation

The need to measure computer security and the difficulty of demonstrating satisfaction of security requirements is well known. Although a variety of security assessment techniques have been developed, the inherent complexity of modern computer systems and the relative lack of experience in performing positive measurements of security have hampered progress towards standardized approaches for measuring levels of computer security. This initial effort is intended to identify issues relevant to performing computer security evaluations and to analyze the characteristics and experience with specific security analysis techniques that have been developed.

This technology assessment, by analyzing and comparing the major approaches and methodologies in use today, forms a logical basis for subsequent efforts to provide guidance to Federal ADP [0] managers in structuring and conducting security evaluations of sensitive computer applications, systems, and installations for purposes of certification and accreditation. Recommendations and guidance on the use of specific or generic evaluation techniques are derived in this technology assessment.

---

[0]Although the Federal government uses the broader term "ADP" (automatic data processing) to describe its computer activities, this document uses "EDP" (electronic data processing) interchangeably because of the usage in the private sector. Two examples of this usage are: EDP Auditors Association and EDP Insurance.

1.2 BASIC TERMINOLOGY

The terminology used in this document is derived from the computer security and audit communities. A number of these basic terms are defined here. Although many of the terms have similar subject matter, they usually reflect the originating community. The security audit definition reflects the traditional auditor's concern with the control of the system and the validity of data processed, while certification focuses on support of management accreditation. Thus, the terms computer security certification and computer security audit are defined somewhat differently, although both processes could be performed using the same techniques. Other terms or phrases such as "measuring the level of computer security" or "security evaluation" are very closely related since a security evaluation must be based on some kind of measure of security level. Based on this relationship, this document treats these two activities as synonymous.

Again, because of the different communities from which many of the security evaluation methods evolved, there is terminology integral to the methodology that reflects the particular community's viewpoint. Thus, methods deriving from the Department of Defense community are concerned with the protection of classified assets whereas auditors are concerned with protection of valuable and sensitive assets. Despite these differences in emphasis, however, the issues are basically similar. As shown in this document, the terminologies can usually be mapped from one arena to another and between techniques. On the other hand, there are subtle distinctions of focus or approach that are characteristic of techniques which have their roots in certain communities. These distinctions are highlighted in the document.

The following definitions of key relevant terms are taken mainly from FIPS PUB 39 [FIP39], and NBS Special Publication 500-57 [RUT80]. The definition of "security requirements" is based on the editor's current understanding. See FIPS PUB 102, Appendix A, [FIP102] for additional relevant terms and discussion of terminology.

1. Accreditation [FIP39] [FIP102]
   The authorization and approval, granted to an ADP system or network to process sensitive data in an operational environment, and made on the basis of a certification by designated technical personnel of the extent to which design and implementation of the system meet pre-specified technical requirements for achieving adequate data security.
   (This document assumes that the definition also applies more broadly to computer security in general, not just data security, and to sensitive computer applications that might not contain sensitive data.)

2. ADP System Security [FIP39]
   All of the technological safeguards and managerial procedures established and applied to computer hardware, software, and data in order to ensure the protection of organizational assets and individual privacy.

3. Audit Trail [FIP39]
   A chronological record of system activities which is sufficient to enable the reconstruction, review, and examination of the sequence of environments and activities surrounding or leading to each event in the path of a transaction from its inception to output of final results.

4. Automated Security Monitoring [FIP39]
   The use of automated procedures to ensure that the security controls implemented within an ADP system are not circumvented.

5. Certification [FIP39] [FIP102]
   The technical evaluation, made as part of and in support of the accreditation process, that establishes the extent to which a particular computer system or network design and implementation meet a prespecified set of security requirements.
   (Since certification is by definition part of the accreditation process, a mandate for certification (e.g., [OMB78]) carries with it an implicit mandate for accreditation. This document uses the terms computer security certification, security certification, and certification synonymously.)

6. Computer Security Audit [RUT80]
   An independent evaluation of the controls employed to ensure:

   (1) the appropriate protection of the organization's information assets (including hardware, software, firmware, and data) from all significant anticipated threats or hazards,

   (2) the accuracy and reliability of the data maintained on or generated by an automated data processing system, and

   (3) the operational reliability and performance assurance for accuracy and timeliness of all components of the automated data processing system.

7. Data Security [FIP39]
   The protection of data from accidental or malicious modification, destruction, or disclosure.

8. External Security Audit [FIP39]
A security audit conducted by an organization independent of the one being audited.

9. Internal Security Audit [FIP39]
A security audit conducted by personnel responsible to the management of the organization being audited.

10. Penetration Testing [FIP39]
The use of special programmer/analyst teams to attempt to penetrate a system for the purpose of identifying any security weaknesses.

11. Personnel Security [FIP39]
The procedures established to insure that all personnel who have access to any sensitive information have the required authorities as well as all appropriate clearances.

12. Physical Security [FIP39]

(1) The use of locks, guards, badges, and similar administrative measures to control access to the computer and related equipment.

(2) The measures required for the protection of the structures housing the computer, related equipment and their contents from damage by accident, fire, and environmental hazards.

13. Risk Analysis [RUT80]
An analysis of an organization's information resources, its existing controls, and its remaining organization and computer system vulnerabilities. It combines the loss potential for each resource or combination of resources with an estimated rate of occurrence to establish a potential level of damage in dollars or other assets.

14. Security Requirements [FIP102]
Identified security needs.
(These needs are expressed in Federal laws and regulations, agency standards and policies, and User's Project Requests. These characteristics are often modified by top management's view of assets and risks.)

15. Testing [FIP101] [FIP102]
Testing, either automated or manual, examines system behavior by executing it on sample data sets.

16. Validation [FIP39]
The performance of tests and evaluations in order to determine compliance with security specifications and requirements.

17. Verification [FIP101]
Verification employs integrity and evolution checking to determine internal consistency and completeness.

1.3 TECHNOLOGY ASSESSMENT OVERVIEW

This section provides an understanding of the scope, organization, and information sources of this document.

1.3.1 Scope

This document covers methods for measuring the level of computer security, i.e., technical tools or processes which can be used to help establish positive indications of security adequacy in computer applications, systems, and installations. The report addresses individual techniques and approaches, as well as broader methodologies which permit the formulation of a composite measure of security that uses the results of these individual techniques and approaches. The unifying concept for the report is security adequacy. This concept influences both the formulation of the security requirements and the determination of acceptance criteria (which express the desired relationship between evaluation results and security requirements) [NEU82].

1.3.2 Organization

The report organization roughly models the security evaluation process: the establishment of requirements and an evaluation approach; followed by the employment of one or more technical evaluation methods; concluding with an interpretation of the evaluation results and a determination of whether or not security is acceptable. This same model can be seen to apply equally well to applications, systems, and installations although specific techniques are often more relevant for certain entities to be evaluated.

In Chapter 2, a discussion of environments and their impact on the formation of security requirements and, to a lesser degree, security evaluation approaches, is presented. Chapter 3 provides descriptions and analysis of major security control groupings. Control groupings frequently serve to orient subsequent evaluations of specific controls. Techniques and methods for evaluating the effectiveness of security controls are presented in Chapter 4. These techniques are divided into two categories: security evaluation methodologies (many derived from the audit community) and risk assessment methodologies. For each methodology, a description of

specific characteristics and features is provided. Chapter 5 summarizes the state of the art of security evaluation. Differences between audits, security evaluations, and risk assessments are analyzed. General evaluation issues of quantification, uncertainty and bias, and integration with the decision process are discussed. Chapter 6 discusses the impact of security policy by defining security sensitivity distinctions and correlating acceptance criteria with the results of specific or collective security evaluations. Discussions in this section illustrate the difficulty of defining precise security measurement criteria and standard sensitivity distinctions. Chapter 7 provides an overview of the document. An Executive Summary of findings and conclusions appears in the prefatory section of this document.

1.3.3 Sources And Approach

In assessing technologies suitable for use in measuring computer security, techniques and concepts fostered in the audit community and the computer security arena, including the Department of Defense, are considered. Useful insights derived from related fields such as forecasting and decision theory are also included where they provide additional insight to subtle problems or issues which are central to security measurement.

A review of major known security evaluation methodologies and approaches has provided the primary source material for this document. A complete list of sources is provided.

After the production of the initial draft of this report it was realized that the Verification, Validation, and Testing (VV&T) activity in the system life-cycle process was also a valid source of security evaluation methodologies whenever specified system requirements were also security requirements. However, VV&T is as yet a very uncommon activity in organizations today. Therefore the discussions comparing methodologies will not include considerations of VV&T. For further information (1) on VV&T, see [FIP101] and (2) on VV&T tools and techniques, see [ADR81] and [POW82].

CHAPTER 2

ENVIRONMENTS

The things, conditions, circumstances, and influences surrounding and affecting the development and operation of a computer installation, system, or application within an organization represent its environment. Environmental factors help to define the nature of both security as a whole and the security evaluation process in particular for entities within an organization. In this chapter a number of these environmental factors are examined.

2.1 ENVIRONMENTAL FACTORS INFLUENCING SYSTEM SECURITY

A direct relationship exists between management policies about security and management policies about security evaluations. The nature of this relationship is that increasing emphasis on security usually results in a corresponding increased emphasis on security evaluations. Similarly, there is a direct relationship between the level of "protection requirements" in a system and the level of detail required in the security evaluation of that system. (Guidance on level of detail in the AFIPS checklist supports this notion [AFI79, p. 11].) These relationships suggest that factors which influence system security requirements can have a corresponding influence on the security evaluation process.

2.1.1 An Early Approach (1977)

There are many environmental factors which influence system security. Formal organizational security policies and requirements; informal organizational policies and mores; system configuration characteristics and functional requirements; processing modes; and information sensitivities are but a few. Little work has been done at the generic level on analyzing the influences of such environmental factors. There was, however, a session on "Audit Considerations in Various System Environments" chaired by Carl Hammer at the 1977 NBS Invitational Workshop on Audit and Evaluation of Computer Security [RUT77, pp. 6-1 - 6-23]. Most of the session concentrated on the parallel roles served by designers and auditors and the systematic use

of checklists to support these roles. The system environment was said to be established by a detailed checklist of system characteristics. The session categorized these environmental characteristics as shown in Table 2-1. The areas under each major category were said to be "typical" or "representative" with "many more" areas required for consideration in an actual audit.

Table 2-1. Environmental Parameter Categories
[RUT77,pp. 6-16 and 6-17]

PHYSICAL	1.	Location (e.g. flood plain; third floor; multiple)
	2.	Survivability (e.g. high; low)
SYSTEM	1.	Degree of sharing (single or multiple user(s))
	2.	Type of service (batch or interactive)
	3.	Organization (centralized or distributed)
	4.	User access (local or remote)
	5.	Applications mix (dedicated or multi-purpose)
ADMINISTRATIVE	1.	Sensitivity
	2.	Postulated threats

These environmental categories were then used to define four illustrative types of application:

1. General purpose or multiuser programming system (e.g. college computing center).

2. Dedicated data base management system (e.g. airline reservation).

3. Distributed multiuser remote access (e.g. Electronic Funds Transfer System (EFTS)).

4. Dedicated batch-dollar disbursement (e.g. welfare system).

For each of these, the applicability and adequacy of different control techniques were examined and importance ratings were given by a consensus of the session attendees, using a rating range of 0 to 10. Table 2-2 shows one of these examples. Unfortunately the product of this informative exercise serves mainly to correlate control techniques with application classes (as directed in the charge to the session). The session did not examine the individual impact of the different environmental factors. The session is noted here primarily for its initial work in defining environmental categories.

2.1.2 A More Detailed Later Approach (1979)

The impact of environmental factors was, however, addressed at the 1979 Summer Study on Air Force Computer Security [AF79, pp. 68-89]. A summer study session on Secure-System Evaluation chaired by Peter Tasker considered application environmental areas and generic application areas for the purpose of relating them to operating system protection levels.

The application environmental areas considered were:

1. Processor coupling (e.g. loosely coupled interfaces restrict the flow of information).

2. User/data exposure (i.e. risk based on matching user clearance and data classification).

3. Developer/user trust (i.e. the degree of trust placed in developers and users).

4. User capability (i.e. buttons, commands, transactions, or programming).

The generic application areas were:

1. Communications processor or communications switch (limited functionality, little or no interaction by a user).

2. Front-end processor (interfaces hosts and terminals to a network; some of its processing can be altered in response to user terminal inputs).

3. Automated message handling (performs routing, indexing, storage, text generation, editing).

4. Stand-alone, interactive data-management system (data management services supporting batch or terminal entries).

Table 2-2. Dedicated Data Base Management System (e.g., Airline Reservations)
[RUT77, p. 6-21]

	ENVIRONMENT	CONTROLS	RATINGS *
PHYSICAL	LOCATION: Multiple SURVIVABILITY: High SPECIAL: Dial-In Access	PERIMETER CONTROLS BACKUP SITES DISPOSAL CONTROLS COMMUNICATIONS PROTECTION	5 / - / 5 - / 3 / 7 4 / - / - 0 / - / 6
SYST EMS	DEGREE OF SHARING: Multiuser TYPE OF SERVICE: Interactive SYSTEM ORGANIZATION: Distributed USER ACCESS: Remote APPLICATIONS MIX: Dedicated	INTERNAL ACCESS CONTROLS PROGRAM INTEGRITY MEASURES ERROR DETECTION/CORRECTION AUDIT TRAILS FAILURE RESPONSE COMMUNICATIONS PROTECTION	7 / - / 4 - / 7 / - - / 5 / - 1 / 6 / - - / 4 / 8 0 / - / 0
ADMINISTRATIVE	TYPE: Sensitive THREATS: Denial of Service Unauthorized Disclosure of Data Remote	PERIMETER ACCESS PROCEDURES MAINTENANCE ACCESS PROCEDURES BACKUP PROCEDURES PERSONNEL PROCEDURES DEVELOPMENT PROCEDURES	4 / - / 4 6 / 6 / 8 - / - / 8 2 / 8 / 5 4 / 7 / 9

* Note: ACCESS CONTROL / ACCURACY / AVAILABILITY

5. Internetted, interactive data-management system (DMS) with command and control (C2) applications (builds upon stand-alone DMS adding other nodes and complex applications).

6. Dedicated, real-time system with sensor input and forced output (dedicated to a specific function, processing well-structured transaction and sensor-system input).

7. Generalized, real-time system with sensor input and generalized output (similar to above with many concurrent applications).

8. Development facility (not actually an application, but an environment to develop others; may restrict access to developmental programs or support concurrent operational processing).

The session's findings were of interest to the computer security community. First, the group determined "that the application areas were not sufficiently defined to show a clear correlation between application areas and the level of protection required" [AF79, p. 84]. Another interpretation, however, also seems plausible to the writers of the present document and that is, that the general protection requirements of a system are not primarily determined by its application area. An inescapable conclusion is that more research is needed in this area.

A second finding of the group was that "user capability and user/data exposure strongly correlated with the levels of protection needed" [AF79, p. 84]. (As a result, security can be increased by either providing additional controls or reducing user capability or exposure [GIL80, p. 2-11].) These characteristics were then used in an experiment to define levels of protection (see Section 4.1.4.1). Processor coupling also was seen to have a direct influence on protection requirements, with tighter coupling requiring more stringent protection.

All of the Summer Study findings support the supposition that a small number of environmental characteristics is sufficient to define the general protection requirements of a system. Should this be true, it would facilitate the high-level categorization of systems based on their key environmental characteristics. This would be very important since it might permit the formulation of more specific security policy guidelines.

Table 2-3. Environmental Influences [*]

		Capability	Assurance
1.	FUNCTIONALITY	a. User capability b. Degree of sharing 1) External (single/multiple users) 2) Internal (interprocess/ interprocessor)	a. Developer objectives b. Developer trust
2.	SENSITIVITY	a. Information b. System	
3.	POLICY	a. Laws b. Standards c. Regulations d. Formal internal policies e. Informal policies f. Due professional care	
4.	PERFORMANCE	Capability a. Response time b. Throughput c. Accuracy d. Availability e. Survivability f. Measurement ability	Assurance a. Developer objectives b. Developer trust
5.	POSTULATED THREATS	System Reliability a. Hardware b. Software c. Communications d. Applications e. Human	Disasters a. Natural b. Financial
			Malicious Acts a. Users b. Operators c. Developers d. Non-users, internal e. External personnel
6.	OTHER	a. Budget constraints b. Personnel capabilities c. Maintainability d. Penetration resistance e. Auditing and monitoring needs	

---

[*]Synthesized from [AF79, AFI79, RUT77] and internal SDC documents.

2.1.3 Conclusions About Environmental Factors And System Security

In summary, environmental influences are factors which affect security requirements. They represent those forces which ultimately cause each system to be unique. As a result, environmental influences are far reaching. Table 2-3 presents a general structure of the influences. It is partially synthesized from the work discussed above but is more encompassing. These influences would have to be considered on an installation, system, and application basis.

Some of the influences deriving from performance needs and postulated threats illustrate that environmental factors influence all forms of potential security exposures (e.g. denial of service), not just data disclosure. The Summer Study session was concerned only with data disclosure. Any high-level categorization derived from key environmental influences should consider all security impact classes.

Finally, much more thought must be given to questions such as the following:

1. Can a small number of key environmental factors be used to consistently place systems into "protection" categories, based on their required "level" of security?

2. Which factors are key? How do the factors differ depending upon the type of security exposures being considered?

3. What uses would such a categorization serve? What dangers or misuses are conceivable?

4. When do combinations of non-critical factors become sensitive? Is there some measure for this?

2.2 ENVIRONMENTAL FACTORS INFLUENCING SECURITY EVALUATION

The major environmental factors influencing both security requirements and the security evaluation are those discussed above. The purpose of this section is to clarify how some specific factors influence the security evaluation process itself.

The level of detail required in the evaluation is strongly influenced by the level of risk and corresponding protection requirements. Influences of the functionality, sensitivity, and postulated threat categories of Table 2-3 seem very relevant here. If the primary security concerns are performance-relevant, the performance category would become critical. Policies, whether internal or external, formal or informal, also play a major role in

determining level of detail. Budget constraints would be another key influence in this area.

The AFIPS checklist [AFI79, p. 11] lists other situations which affect level of detail:

- In a center which has never addressed security, the initial evaluation could be simple with an in-depth review reserved until minimum standards have been met.
- Prior unfavorable evaluations might impose requirements for added detail.
- Critical tasks would require a more detailed evaluation than tasks which weren't critical.

It also notes that "costs tend to increase at a decreasing rate as the size of the installation increases....Larger systems tend to complicate the review up to a certain point, and then the size becomes less influential on the cost of the review" [AFI79, p. 11].

Similar factors can influence the frequency and scope of the evaluation in addition to the level of detail. Also, security-relevant changes to policies, assets, threats, or controls might require immediate evaluation. The technical architecture involved may impact the detailed content of the evaluation.[1] Detailed content will also be strongly influenced by whether the evaluation focus is on installation, system, or application concerns and by the phases of the development cycle which are involved.

Environmental factors, then, clearly impact the nature of the security evaluation process. Many of the impacts, however, involve issues of level of detail, frequency, and scope which may be more significant because they can strongly impact the certification program in which the security evaluation is embedded.

---

[1]It is commonly accepted in the audit community that different architectural environments require different control and audit techniques. One audit source [CIC75, pp. 256-258] lists four such environments as those employing:

1. Remote terminals (being used to access and update on-line files).
2. Multiple CPUs (which share files).
3. Data base management systems.
4. Distributed processing systems.

2.3 CONCLUSIONS

Environmental factors include everything that influences a system. Some factors are more important than others in determining security and security evaluation requirements. Potentially the most fruitful area needing research is that of determining the extent to which a small number of environmental factors can be used to define and categorize systems. Little work has been done in this area. Environmental influences greatly determine the nature of an organization-specific evaluation methodology.

CHAPTER 3

CONTROLS

This section examines alternative ways to classify and group ADP security controls. Because of the great variety of individual controls, and the extensive treatment of these controls in other places, examination is focused primarily on classes and roles of security controls. Exposure groupings are also examined because the exposure structure describes the security problem that the security controls are attempting to solve. Emphasis is placed on the influence of these groupings or structures upon security evaluation. An underlying characteristic of this analysis is that the quality of one's groupings reflects the quality of one's understanding of the security problem and its solution. Note that the evaluation methods that use these structures are discussed in Chapters 4 and 5.

3.1 CONTROL GROUPINGS OR STRUCTURES

Control groupings from the following primary references were examined and analyzed:

- Systems Auditability and Control Study, SRI [IIA77-2]
- Security: Checklist for Computer Center Self-Audits, Browne [AFI79]
- Army Regulation 380-380 [USA79]
- Computer Control Guidelines, CICA [CIC70]
- Information Security Handbook, IIA [WIL80]
- Control Objectives 1980, EDP Auditors Foundation [EAF80] [2]
- Course in internal controls for auditors offered by GAO [GAO80]
- NBS SP 500-57 [RUT80]
- EDP Insurance [MIG80]

---

[2]The EDP Auditors Foundation updated this document in 1983. Since the control groupings were changed somewhat in the newer version, this document will reference the 1980 version in the interest of expediency. Since there are no generic preferred groupings, this will not alter the validity of the discussions.

3.1.1 Systems Auditability And Control (SAC) Study [IIA77-2]

The SAC report divides controls into three categories:

- Applications systems controls
- Computer service center controls
- Application system development controls

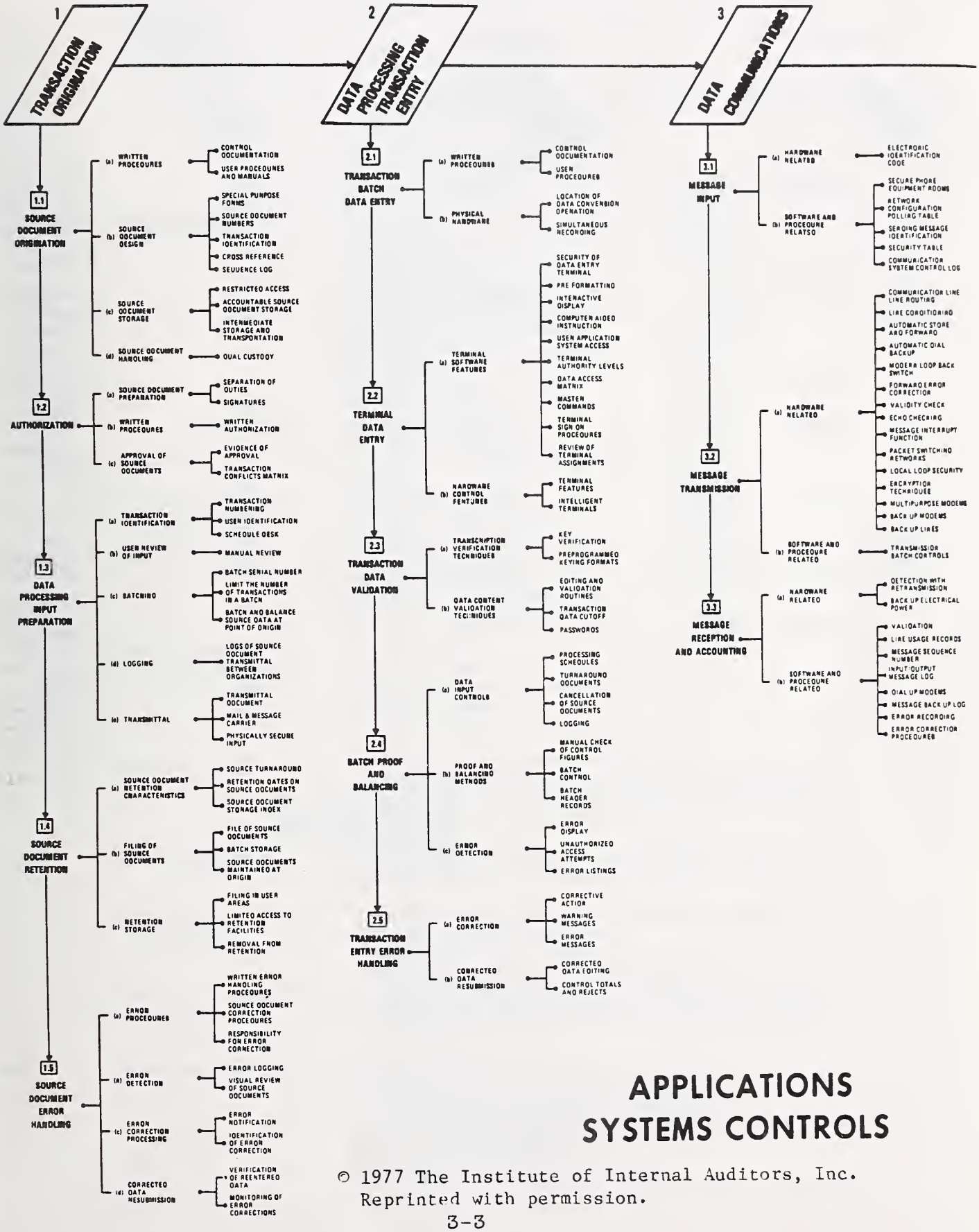
A separate set of groupings is provided for each. The applications systems control structure is by far the most detailed and is oriented around a transaction flow approach with the high-level groupings consisting of transaction origination, data processing transaction entry, data communications, computer processing, data storage and retrieval, and output processing. Each of these is then further divided into transaction oriented subgroupings (see Table 3-1). The computer service center controls are organized into installation oriented concerns (see Table 3-2). Application system development controls emphasize change control, documentation, and especially review during different phases of the system development life cycle (see Table 3-3).

The application groupings seem very comprehensive. The use of a transaction flow approach to application controls is consistent with current practice in the accounting community for evaluating applications [AAC78]. Of course financial processing by its nature tends to be oriented around transactions. However, not all applications are financial or transaction oriented[3]. Some may be session oriented as in interactive on-line development or in communication. Whatever the orientation, work units of some sort exist and the basic input-process-output functionality of the SAC groupings is applicable. The further an application differs from an accounting oriented process, however, the more the lower level application control structure will differ.

---

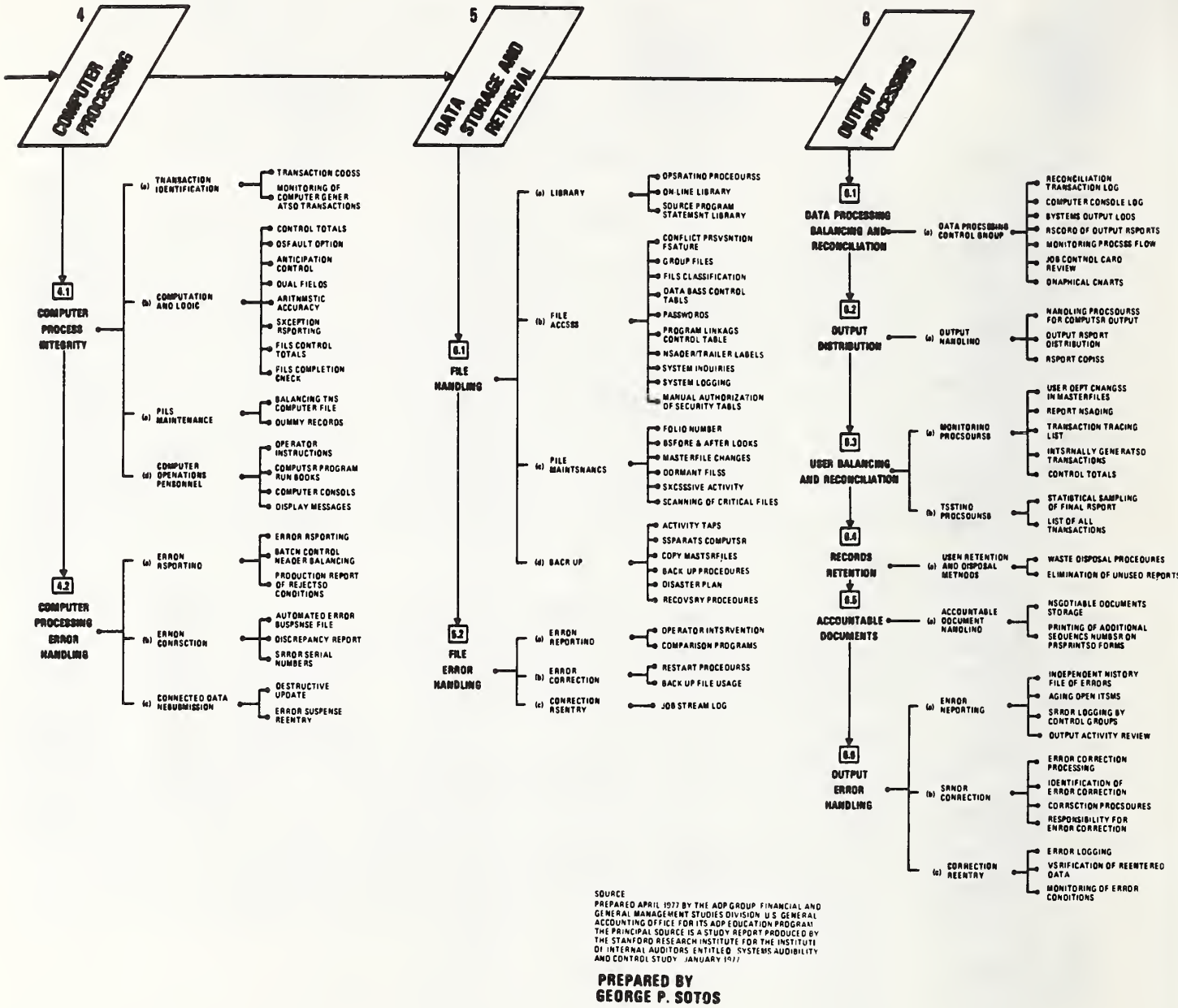
[3] One government agency defines two types of environment, static and dynamic. A static computational environment includes more structured processing with limited time sharing, such as process control systems or transaction oriented systems supporting one or two applications. A dynamic environment includes more diverse processing using multiple partitions for tasks such as scientific computation [NRC80].

Table 3-1. Applications Systems Controls from the Systems Auditability and Control Study [IIA77-2]



APPLICATIONS SYSTEMS CONTROLS

Table 3-1. Applications Systems Controls from the Systems Auditability and Control Study [IIA77-2] (Cont'd)



SOURCE
PREPARED APRIL 1977 BY THE AOP GROUP FINANCIAL AND GENERAL MANAGEMENT STUDIES DIVISION U S GENERAL ACCOUNTING OFFICE FOR ITS ADP EDUCATION PROGRAM THE PRINCIPAL SOURCE IS A STUDY REPORT PRODUCED BY THE STANFORD RESEARCH INSTITUTE FOR THE INSTITUTE OF INTERNAL AUDITORS ENTITLED SYSTEMS AUDIBILITY AND CONTROL STUDY JANUARY 1977

PREPARED BY
GEORGE P. SOTOS

Table 3-2 **COMPUTER SERVICE CENTER CONTROL STRUCTURE** [IIA77-2, p. 91]

Control Area	Control
Input/output scheduling and control	Input/output control group Job handling procedures Scheduling procedures Processing schedules Cutoff dates Job submission/authorization Input/output control group Daily work schedules
Media library controls	Access controls File release Separate library area Physical security Media restriction issuance Media inventory control procedures Media restriction quantity Media use records Handling equipment Security vaults Temperature and humidity control Off-site storage Duplicate file storage Redundancy and backup procedures
Malfunction reporting and preventive maintenance	Procedures reporting Problem documentation System utilization report System utilization reports review Vendor failures logging Resolutions problem logging Trouble reporting responsibility
Environmental controls and physical security	Physical security Access controls Backup for power, cooling, etc. Software backup Hardware backup
Separation of duties	Separation of duties within data processing Separation of data processing from other organizational units Computer programs to enforce separation between systems Automated controls to separate on-line users
Resources planning	Plan for facilities, equipment, software, and personnel Variance between actual and planned goals
User billing/charge-out procedures	Service contracts between the user and data processing Procedures to arbitrate disputes Chargeable versus free services Billing procedures tied to the computerized job accounting system Billing algorithm, periodic user billing statements, rerun cost allocation procedure
Disaster recovery	Disaster plan Disaster scenarios to update the disaster plan Top management's commitment to the disaster plan Maintenance and updating of the disaster plan User responsibility for the disaster plan Testing of the disaster plan User training in use of the disaster plan

Table 3-3

APPLICATION SYSTEM DEVELOPMENT CONTROL STRUCTURE [IIA77-2, p. 10.]

Control Area	Control
System development life cycle	Review user requirements Review project organization Review hardware requirements Review internal controls Require user or internal auditor sign-off Review detailed design documents Review file requirements Review costs and schedules Review test plan Review user and operational documentation Review test results Require user and internal audit sign-off Review conversion plan Review adequacy of documentation Identify user problem areas Identify system development problems
Project management	PERT control technique
Structured programming	Audit trail
Acceptance testing	Comparison of base case data to system produced data Manual review of output reports
Program change control	Require formal written request Review all changes Use of change control committee Restrict number and type of persons who make changes Require operator and programmer to make changes Modify SMF to obtain reports of all changes to load libraries Require report of quick fixes Limit number of times changes are made per time period Use program packages to control access to source libraries
Documentation	Provide flow of all application system data flow Specify how the programs implement controls Specify how programs are to be operated, backed up, and recovered Maintain up-to-date changes to accepted documents Specify allowable user commands and functions
Data base administration	Develop standards Establish and monitor standards Document and provide procedures to control operations Measure effectiveness of performance and integrity controls Develop security/control education programs

Whereas the applications and service center groupings emphasize control functions required in running a system, the development structure emphasizes control functions required in building or changing a system. These developmental control functions contribute to the quality of other controls and thus can be seen as quality assurance functions. Such quality assurance functions might not have to be treated explicitly, since they could be viewed as implictly represented in the controls developed from them. From the point of view of an auditor or evaluator, however, quality assurance steps seem best treated as explicit, auditable control functions, because of their importance in influencing control quality. In other words, the role of the auditor is to evaluate the entire process, which includes both developing and operating the controls.

The service center controls do not address many organizational and policy issues. Also they do not separately address architectural and system hardware and software issues which seem best considered apart from application controls.

3.1.2 Security: Checklist For Computer Center Self-Audits [AFI79]

The AFIPS security checklist is divided into nine categories as shown in Table 3-4. This structure provides much insight into the full extent of security controls. Particularly interesting is its segregation of program and management issues, such as Planning and Risk Analysis, and The Security Audit into separate categories. The structuring under Administrative Controls which includes Policy, Laws, and Insurance, as well as Data Entry and Output Handling, is similarly attractive since it should correlate well with organizational responsibilities. The subdivision of Physical Access Controls into Perimeter, Building, Sensitive Office Area, and Computer Area is useful and exemplifies how a simple, insightful structure might clarify analysis.

This AFIPS document's treatment of applications controls and the development process differs markedly from the SAC [IIA77-2] approach. Developmental phases (rather than a transaction flow sequence) are used to structure application controls. A primary reason for this is probably that audits have tended to emphasize review of applications which are already operational, whereas the AFIPS checklist, while not excluding operational reviews, places particular emphasis on reviews during development. This different emphasis results in a redundant analysis of development controls (i.e. controls examined during developmental phase reviews) as opposed to a centralized analysis of the development process as a whole. Primary redundancy occurs in the areas of Systems Hardware and Software, Communications, Applications, and The Security Audit. This is not necessarily a disadvantage, because although the AFIPS structure may increase redundancy, it allows tailoring of detailed design review questions to the somewhat different issues involved in design review for, say, Operating Systems and Applications.

Table 3-4. AFIPS Guideline Evaluation Categories [AFI79]

Planning and Risk Analysis
- Security Program Development and Administration
- Risk Analysis

Physical Security
- Access Controls
  - o Perimeter
  - o Building
  - o Sensitive Office Area
  - o Computer Area
- Protection of Sensitive Information
- Fire Prevention and Detection
- Electrical Power
- Environmental
- Computer Facility

Backup and Recovery
- Short-term Recovery
- Off-site Backup
- Disaster Planning
- Disaster Recovery

Administrative Controls
- Personnel
- Security Standards and Procedures
- Legal Aspects
- Data Entry
- Output Handling
- Insurance

Systems Hardware and Software
- Operating System Characteristics
- Modifications by Enhancements
- Operations, Maintenance, and Generation

Communications
- Communication and Switching Software
- Encryption
- Logs and Audit Trails
- Physical and Procedural Controls
- Planning and Design of Communication Networks

Distributed Risk
- Stand-alone Small Systems
- Remote Terminal Controls
- Remote Processing

Table 3-4. AFIPS Guideline Evaluation Categories [AFI79] (Cont'd)

Applications(integrity included)
- Design Considerations
- Development Process
- Implementation Process
- Operational Considerations

The Security Audit
- System Auditability
- Audit Data
- Audit Tools and Techniques

This redundancy might influence detailed checklists to adopt different structures from more general control categorizations. That is, from the above case, a high-level structure can simply include the phases of design review (as the SAC study does) under one system development category. A checklist, however, must recognize that the questions asked during this design review will differ depending on which system component is being reviewed and thus could force inclusion of these questions in several categories, rather than one.

Other characteristics of the checklist structure are that the data base is included under consideration of the operating system, not treated separately. Malfunction reporting and preventive maintenance are also not treated separately. This structure seems to reflect a slight orientation toward components rather than functions. The existence of a separate category for systems hardware and software also illustrates this. Audit community control groupings tend to stress a somewhat more function-oriented structure.

3.1.3 Army Regulation 380-380 [USA79]

Army Regulation (AR) 380-380 [4] is one of the early Federal documents that called for a management component in security evaluation. It uses two different control structures for scoping the tasks. One, essentially conveyed by the table of contents, is as follows:

---

[4] An adaptation of AR 380-380 to permit wider use is available in [CAM80].

1. U.S. Army Automation Security Program
2. Physical Security
3. Personnel Security and Surety
4. Communications Security
5. Emanations Security
6. Hardware Security
7. Software Security
8. Procedural Security
9. Risk Management
10. Accreditation

The other is the structure for a security checklist:

1. Security Management
2. Physical Facilities
3. Personnel
4. Hardware
5. Software Security
6. Service Personnel
7. Files
8. Internal Audit Controls
9. Time-Resource Sharing
10. Contingency Plan
11. Use of Service Bureaus

The primary reason for the two different sets of groupings would seem to be the different structures used for policy and operation. The first set of groupings is used to summarize policy and is therefore based on the structure inherent in Army policy organization and documentation. The checklist, on the other hand, is based on much lower level organizational and operational structures. For example, communications and emanations security, treated separately in the "policy" structure, are included under hardware in the checklist, with both likely to fall under the purview of a single "facilities" manager.

Technical areas such as files, internal audit controls, time-resource sharing, and use of service bureaus are segregated in the checklist while not mentioned in the overall policy structure. Separate treatment is also given to service personnel and a contingency plan in the checklist structure. The existence of these separate lower-level categories in the checklist reflects the difficulty of correlating them with the higher-level policy categories. It should be noted that the Army recently revised this document.

Table 3-5. Control Evaluation Guide of the Canadian Institute of Chartered Accountants [CIC75, p. 262]

Control Evaluation Guide

Summary

Control objectives		Summary of evaluation and major recommendations
I PRE-INSTALLATION	A Benefits of processing alternatives B Selection of facilities C Pre-installation plan	
II ORGANIZATIONAL	D Segregation of functions E Deployment of resources	
III DEVELOPMENT	F Benefits of processing alternatives G Development of effective systems and programs H Maintenance of systems and programs	
IV OPERATIONS	I Prevention or detection of accidental errors J Prevention or detection of fraudulent manipulation K Security against accidental destruction	
V PROCESSING	L Completeness of data M Accuracy of data N Authorization of data O Adequacy of management trails	
VI DOCUMENTATION	P Existence of adequate documentation Q Systems documentation R Program documentation S Operating and user instructions	

3.1.4 Computer Control Guidelines [CIC70]

Table 3-5 shows the control objective groupings used in this CICA guideline. The high-level structure is clearly much more function oriented than the security checklists described in Section 3.1.3. The focus of the approach is the "point of incidence of the error to be prevented" rather than the "point of incidence of the control".

> "If the controls were reviewed according to the point of incidence of the control (e.g. 'input controls', 'programmed controls', 'output controls', etc.) it would be difficult for the reviewer to take these various alternatives into account. On the other hand, if the controls are reviewed according to the point of incidence of the error to be prevented (e.g. 'there should be some method of ensuring that corrections are re-entered into the system for all identified errors'), then the several alternative control techniques which might satisfy this purpose can be listed to enable the reviewer to ensure that at least one of the alternative techniques is, in fact, in force" [CIC70, p. 1].

3.1.5 Information Security Handbook [WIL80]

The primary structural breakdown in this document is a set of checklists as shown below:

1. Information-Security Organization
2. Classification of Information
3. Selective Protection of Information
4. Traditional Security
5. Islands of Security
6. Information Service Functions
7. Top-Priority Document Control
8. New Product Security
9. Trade Secrets
10. Intellectual Security
11. Information Destruction
12. DP Organizational Controls
13. User/Owner Responsibilities in a Data Processing Environment
14. Computer Operations
15. Bulk Data Transmission
16. Remote Computing
17. New System Design, Development, Test, and Implementation
18. Auditability
19. Encryption
20. Top-Priority Data

The central viewpoint is that information is a key asset in an organization and it needs protection. The above categories are oriented around information security issues and functions. The checklists are not oriented around components because the document serves as a general guideline, not as a detailed review aid.

3.1.6 Control Objectives 1980 [EAF80]

This document [5] divides control objectives into four categories: management controls, technical services, applications, and operations. The four categories are further subdivided as follows:

1. Management controls
   - External requirements
   - Planning
   - Organization
   - Policies, standards, procedures
   - Resource Management

2. Technical services
   - Systems programming activities
   - Data base management systems
   - Hardware selection, security, and control
   - System software selection, security, and control

3. Applications
   - Design and development of application systems
   - Programming
   - System validation
   - Implementation
   - Program and system change control
   - Security for systems and programs

4. Operations
   - Scheduling
   - Processing
   - Data Storage
   - Security
   - Computer backup
   - Timesharing
   - Distributed processing

---

[5]See footnote 2.

The high-level structure seems to be primarily organizational, reflecting the fact that many organizations structure their computer-related activities in a similar way. This permits tailoring of the objectives to the different organizational groups. It also results, however, in a fairly high level of redundancy. For example, change controls are addressed in three out of the four major categories (i.e. not operations) and three of the technical services subcategories (i.e. not hardware) contain change-related controls. As another example, standards are addressed under both management controls and the programming subcategory of applications controls. Interestingly, the structure within applications controls is, like the AFIPS checklist, oriented around the development cycle.

3.1.7 GAO Internal Controls Course [GAO80]

A General Accounting Office (GAO) course was given in 1980 and included the following control structure:

1. General controls
   a. Organizational
   b. System development
   c. Data center management
   d. Data center security (including physical controls and contingency/backup)
2. Application controls
   a. Data origination
   b. Data entry/validation
   c. Data communication
   d. Computer processing
   e. Data base controls, maintenance, and recovery
   f. Processing control in advanced systems
   g. Output processing

The application controls category seems to be based on the SAC study [IIA77] application structure, although item "f" has been added. If this is in fact based on the SAC study, it is significant that system development controls have been included under general controls rather than broken out in a third high-level category as in the SAC study (and the Touche Ross methodology discussed in Chapter 4).

3.1.8 NBS SP 500-57 [RUT80]

At the NBS Invitational Workshop on Audit and Evaluation of Computer Security II, documented in NBS SP 500-57, working sessions were structured according to how management responsibility for controls would be assigned to different organizational units. The model organization chart is shown in Figure 3-1. The chart was adapted from NBS SP 500-25 [RUD78] which contains another control listing (of 132 detection and 56 prevention safeguards).

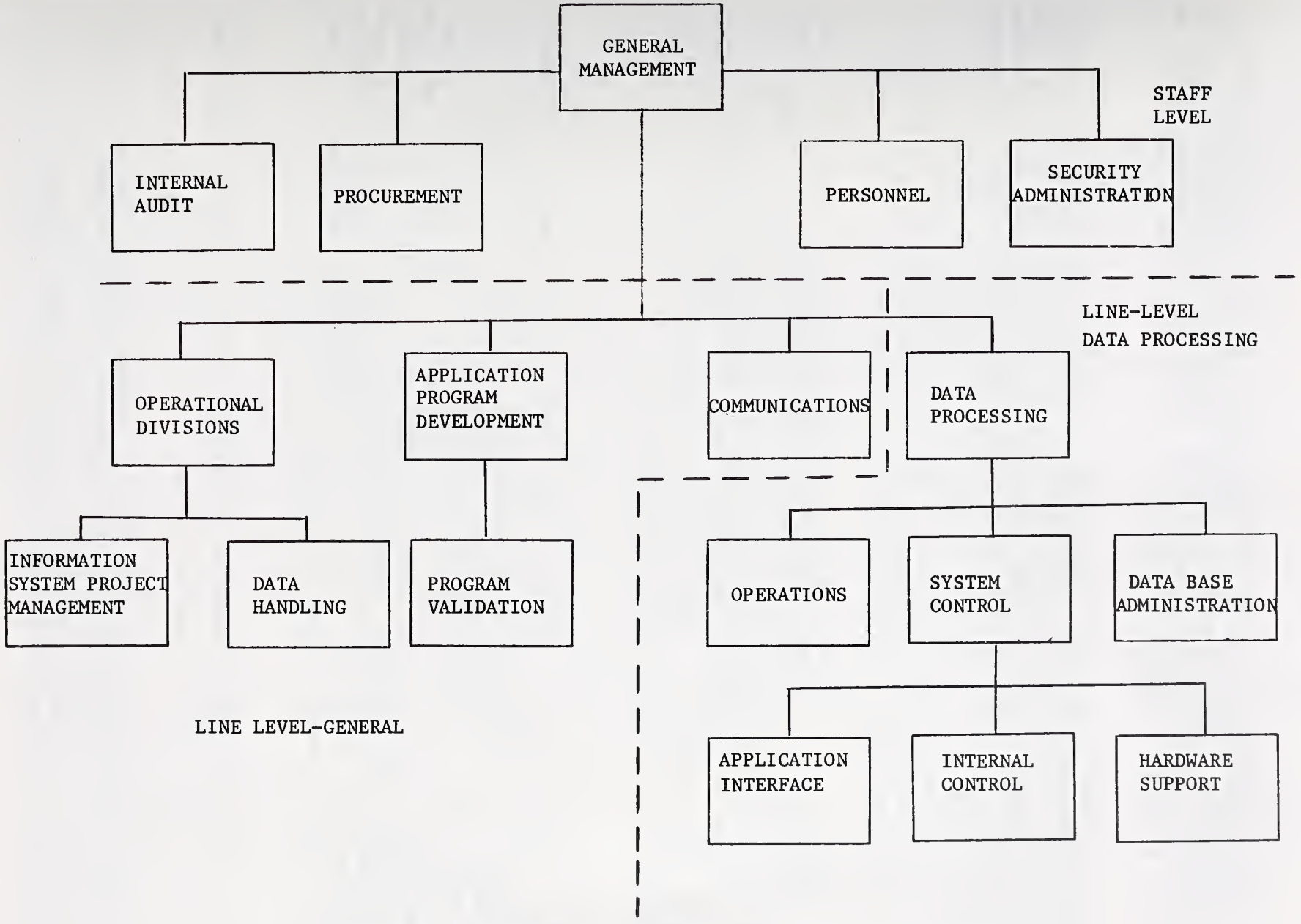


Figure 3-1. A Model for Categorizing System Vulnerabilities and Managerial Controls According to Responsible Organizational Units [RUT80, p. 1-5]

It may be useful to key control structures for evaluation around the organizational structure because this facilitates both partitioning of the work and reporting of results. This fact would be relevant whether the evaluation is being done internally or by an external group for higher management.

One of the sessions at the workshop (session 6) proposed still another control structure yielding a possible metric in evaluating security [6]. This is discussed at length in Section 4.1.4.1 and shown in Figure 4-3. The most notable aspect of the structure is its division of controls into protection mechanisms and assurance features.

3.1.9 EDP Insurance [MIG80]

Insurance [7] provides a form of protection from security risks and therefore is often considered equivalent to a control. Since insurance rates and programs are typically based on empirical data, it is instructive to examine the structure which has developed in the EDP insurance business. There are two main categories of insurance which are relevant in an EDP environment: EDP All Risk and Fidelity. They are kept very separate. EDP All Risk is oriented around the physical environment; Fidelity around employee fraud. Fidelity is not solely an EDP insurance, being much more general in its intended scope.

3.1.9.1 EDP All Risk - EDP All Risk includes equipment and media damage, business loss, and down-time. It is like disaster insurance. It considers such factors as city services involved (e.g. public and volunteer fire protection, water sources) and building codes (e.g. fire resistance). These factors are computed into a "base rate" based on figures from such rate-making bodies as the Insurance Service Office (ISO) which is a non-profit organization. This base rate facet of insurance is heavily regulated. Once the base rate has been established, the underwriter factors in surcharges or credits based on company-unique characteristics (e.g. management attitude, other tenants, use of smoke detectors) [MIG73].

3.1.9.2 EDP Fidelity - Fidelity insurance involves factors such as

---

[6]It should be noted that this session's paper led to the development of the DoD trusted computer system evaluation document [DOD83].

[7] It should be noted that Federal computer systems cannot be insured by the private sector but is self-insuring.

employee stealing and trade theft. It is completely separate from EDP All Risk and is quite judgmental from a rates perspective. Factors affecting rates include the number of employees, cash exposure, internal controls (e.g. who authorizes), and so forth. (These are factors used by Seaboard Surety for their Fidelity rates.)

3.1.9.3 Criteria For Insurance Selection - The primary distinction between the two categories is that EDP All Risk covers natural and environmental threats against physical assets and business services whereas Fidelity covers employee threats against information and monetary assets. The existence of this basic distinction must be kept in mind when structuring a security evaluation. Especially in performing risk assessments, it may be necessary to "total" the risks in each area in order to determine how much of each type of insurance to obtain (since insurance is a countermeasure).

3.2 SUMMARY OF CONTROL GROUPINGS OR STRUCTURES

This section discusses the roles served by control groupings and presents an illustrative general control structure[8].

3.2.1 Roles Served By Control Groupings Or Structures

Control structures serve many important roles in the security evaluation process. The major roles are to:

1. Support or provide overall partitioning of the evaluation analysis.
2. Support the objectives of the evaluation.
3. Support the philosophy or approach of the evaluation analysis.
4. Permit focusing the skills of technical specialists.
5. Correlate with organizational responsibilities, structures, policies, and documents.
6. Prioritize analysis.
7. Clarify the purpose of controls.
8. Clarify the overall control problem.

These are explained further in the following material.

1. Support Partitioning. Because of the breadth and complexity

---

[8] The term "control groupings or structures" refers to the groupings or structures used for both generic control listings and organization specific tailored listings.

of security evaluation, it is critical to partition the entity being evaluated into manageable evaluation components. The control structure used can provide this partitioning. If the partitioning has been provided by some other means (e.g. organization structure), the control structure should support the partitioning. The discussion below notes a number of factors which can motivate different partition structures.

2. Support Objectives. The control structures used must support the objectives of the evaluation. For example, if the primary objective of the evaluation were to assess compliance with a detailed listing of control objectives (as in many audits), the structure will derive from the structure of the control objectives. This might greatly complicate the use of existing generic checklists since objectives tend to be function-oriented and checklists more component-oriented (see Section 3.1.2). Evaluation objectives can vary widely in terms of both level of detail and organizational components, threats, assets, and exposures involved. All of these factors can influence the control structures (including level of detail, which as noted in Section 3.1.2 seems to force a slightly different structure on checklists). As noted in Section 3.1.9, the need to procure different types of insurance may even result in a partitioning of controls according to relevant insurance categories. The impact of different exposures is quite significant and is discussed at length in Section 3.3.

3. Support Philosophy. The philosophy or approach of the evaluation analysis must be supported by the control groupings. For example, selection of a transaction flow approach towards analysis might argue for the use of a transaction-flow-based control structure, such as the SAC study structure for applications controls.

4. Permit Focusing. Many areas of security can be acceptably evaluated only by skilled experts who specialize in those areas. The partitioning described above must support the efficient use of such people by allowing them to focus on their own areas. In some cases, the advice of these experts will be required in order to do the partitioning. For example, it is common to isolate encryption as a subject of relevance only to communication lines. However, file and password encryption are relevant to central processing and storage. In addition, the nature of controls required in the communication protocols above the encryption layer are determined by those present at the encryption layer (and vice versa). This example also serves to illustrate that security controls, like communication protocols, are layered, often with seemingly redundant controls at application, system, and hardware levels.

5. Correlate With Organizational Factors. Control structures must often correlate with organizational responsibilities, structures, policies, and documents. This will allow the reporting of findings to responsible authorities in such a way that individuals are told of shortcomings only in their areas. For example, user departments would be informed of problems only in their applications. It also allows assessment of compliance with policies, which may differ for different organizational components. Finally, findings may require changes to specific policy documents, which would be simplified if a correlation were feasible.

6. Prioritize Analysis. Control structures can be used to prioritize analysis. Touche Ross & Co. divide their controls into prevent, detect, and correct categories and place highest priority on detect controls (since they are auditors)[9] [MAI76, p. 184]. DoD usually places first emphasis on those controls preventing disclosures (as opposed to integrity or denial of service violations)[10]. A list of disclosure-relevant controls is provided in [RUD78].

7. Clarify Purpose. Structures can clarify the purpose of controls. Two examples of structures which do this are given in the previous paragraph. Another example is the familiar access-control, authorization, authentication, audit categorization (which is sometimes expanded to support additional categories such as flow control, inference control, and encryption). Still another example is the segregation of controls into enforcement and assurance mechanisms. For example, system development controls [11] (e.g. structured walk-throughs, use of higher level languages) in a sense address the subject of assurance. This can reflect on the quality (i.e. vulnerability) of each control developed under the system development methodology. As a result, the assessment of an expert on the quality of

---

[9] There is a compelling reason to place major emphasis on detect controls. One point which seems to be emerging from Donn Parker's research on computer criminals is that an embezzler is often not deterred even by quite sophisticated controls, which might in fact serve as more of a challenge than a deterrent. His findings are that "the main deterrent has been the fear of detection and disclosure" [EDP80, p. 6.36]. (Of course detect controls must anticipate perpetrators claiming their activities to have been accidental.)

[10] Integrity violations involve unauthorized modification, e.g., adding, deleting, altering, substituting, or duplicating transactions. Denial of service violations involve users interfering with or inhibiting the provision of service to other users.

[11] For an excellent discussion of development controls, see [BRA78].

the development methodology will reflect on the quality of many controls.

8. Clarify Problem. Finally, control structures can clarify the nature of the overall control problem. Touche Ross uses several structures for this purpose [MAI76, pp. 35-36]: logical versus technical and vertical versus horizontal. Logical controls derive from pre-computer era business logic or practice (and may be implemented in a computer). Technical controls (e.g. parity checks) are new and peculiar to computers. This structure "relates controls to the relative degree of technical education that is found among business managers and auditors". In the second structure, vertical controls follow the vertical lines of authority (e.g. supervision, segregation of duties). Horizontal controls cut across lines of authority (e.g. transmittals between departments) and illustrate the "upward shift in the lowest level of common supervisory or line management control" in ADP systems. This structure reveals the need to place greater emphasis on horizontal controls in ADP systems than in non-ADP systems.

In summary, the selection of one or more control structure(s) influences all phases of a security evaluation. Consideration of control structures is thus critical to the effective, efficient performance of a security evaluation.

3.2.2 General Control Structure

There is no general control structure or grouping which meets all needs. While all the structures examined have strengths and weaknesses, all seem reasonable in terms of meeting their specific objectives. To the extent that there might be value in synthesizing a general high-level security control structure based on the above analysis, such a structure is presented in Table 3-6. However, the most important point derivable from this analysis, is that it would be dangerous to blindly adopt such a general structure. Alternative structures must be thoroughly analyzed and an organization and situation-specific structure adopted to meet the needs of a particular evaluation.

Table 3-6. Illustrative General Control Structure

Installation Controls

- o General organization (resource plans, separation of duties, personnel, administrative, laws, regulations, insurance, security standards)
- o Data entry and output handling (includes scheduling)
- o Data communication (network/system)
- o Physical security and environmental controls
- o Remote terminals and distributed systems
- o Data base (includes library control)
- o System hardware and software
- o Malfunction reporting and preventive maintenance
- o Disaster recovery (both short-term and off-site)

Development Controls

- o Same as SAC study (see Table 3-3) with particular emphasis on the system development life cycle, program change control, and documentation.

Application Controls

- o Same as SAC study high-level structure (see Table 3-1), increasingly varied at lower levels.

3.3 EXPOSURE GROUPINGS OR STRUCTURES

While the above analysis of control structures or groupings was being performed, it became apparent that the different exposure structures or groupings are also of major significance in security evaluation. The reason for this is that the exposure structure is a representation of the security problem while the control structure represents a solution to that problem. This is an important finding of this technology assessment. The following analysis of exposure structures has therefore been included.

There are two distinct structures for exposures (i.e. impacts resulting from loss). These are: (1)to the first order, "mutually exclusive" (which might be thought of as technically oriented) and (2)"overlapping" (management oriented). The adoption of one structure as opposed to the other may be critical in meeting evaluation objectives.

Mutually exclusive exposures are typically used in risk assessments. Table 3-7 shows some variations. Mutually exclusive exposure categories are used so that non-redundant costs can be associated with each type of loss. This permits an Annual Loss Estimate (ALE) to be formulated for the system as a whole when a risk analysis is performed.

Overlapping exposure groupings are typically used in audits. Table 3-8 provides examples. These clearly contain overlapping categories. In the Touche Ross categorization, for example, loss or destruction of assets could cause business interruption, erroneous decisions, statutory sanctions, or other exposures.

In comparing these two groupings it was found that the overlapping exposures include their causes and types of loss as well as effects of loss whereas mutually exclusive exposures include only types of loss. This overlapping structure does not readily permit the overall quantification of losses due to exposures while a mutually exclusive structure does.

Table 3-7. Mutually Exclusive Exposures

FIPS PUB 65 [FIP65]	U.S. Department of Agriculture ADP Security Handbook [DOA77]
o modification of data	o destruction
o destruction of data	o disclosure
o confidentiality of data (unauthorized disclosure)	o modification (fraud)
o processing availability (denial of service)	o availability (denial of service)

SDC RAM [SDC79]	NBS SP 500-19 Sess.6 [RUT77][*]
o destruction	o cost of implementation, development, and operation
o unauthorized disclosure	o effectiveness for access control
o modification	o effectiveness for accuracy
o denial of service	o effectiveness for availability

[*] The session referred to an organization's score in the areas of access, control, accuracy, and availability as its "AAA rating".

Table 3-8. Overlapping Exposures

Touche Ross Co. [MAI76, p.260]

Application Exposures (if project implemented)
- o Erroneous record keeping
- o Unacceptable accounting
- o Business interruption
- o Erroneous management decisions
- o Fraud
- o Statutory sanctions
- o Excessive costs/deficient revenues
- o Loss or destruction of assets
- o Competitive disadvantage

Project Exposures
- o Erroneous management decisions
- o Excessive costs
- o Competitive disadvantage
- o Business interruption (delay timetable)

GAO Internal Controls Course [GAO80]

- o Inadequate execution of operational activities
- o Inadequate decisions based on erroneous data
- o Lack of confidence in the information system
- o Potential organizational interruptions
- o Information too bad to use

The primary and crucial difference between using the two structures, however, is that overlapping exposures can be much more meaningful to management. This is a key point. Since evaluations are done for management, it is critical to report findings in terms that management can best understand and use. For example, suppose an evaluation suggests the need for greatly increased controls. It should not be assumed that the increased security expenditures necessary to acquire added controls will be automatically forthcoming. The decision to spend more money for security must be made by management, with security needs competing against other organizational needs. The most persuasive and appropriate way to present evaluation findings, then, is in those terms most relevant to management and organizational concerns, since this permits management to make the best decision[12].

The nature of the exposure structure to be used (which really defines the security problem) must be determined with management participation at the beginning of the evaluation. Unless this is done, much of the evaluation effort may be wasted because it may then be extremely difficult (or impossible) to map findings based on one structure to the other structure[13]. This is well illustrated in the example above where the mutually exclusive category "destruction of data" can, depending on the situation, cause exposures in practically every one of the Touche Ross exposure categories. Only a reexamination of all situations involving destruction of data could determine which situations were relevant to each of the overlapping categories.

Improper definition of exposure structures is therefore a likely pitfall in the use of established security, risk assessment, or audit methodologies for purposes other than those originally intended. Exposure structures, then, are very important in tailoring a security evaluation process to meet management objectives.

---

[12] Users will also have to be convinced that added controls are justified, since they will bear the brunt of any increased procedural steps and decreased performance resulting from the controls. In fact, user dissatisfaction from increased controls is an example of an "overlapping" exposure which should be considered by management in its decision of whether to acquire the controls.

[13] A possible compromise solution might be the use of a mutually exclusive structure which is more meaningful to management. One candidate is a structure used by Parker which recognizes four kinds of loss: vandalism, information or property fraud and theft, financial fraud and theft, and unauthorized use or sale of services [PAR80, p. 6.46].

CHAPTER 4

SYSTEM EVALUATION METHODOLOGIES

Existing security evaluation techniques are examined in this section. For the purposes of examination, these methodologies have been divided into two groups, security evaluation and risk assessment. Security evaluation methodologies include approaches used by security safeguard evaluation personnel (e.g., security officers) and the EDP audit community, since both groups use similar approaches. Risk assessment methodologies are described separately since this class of security evaluation approaches play a unique role in the security of sensitive applications. Risk analysis, when performed at the initiation of the system life cycle, can be used to define security requirements; when performed in the development phase of the life cycle can be used to validate the security requirements; and when performed during the operations and maintenance phase of the life cycle, can be used to measure the current level of security of the system. As mentioned in Section 1.3.3, the tools and techniques of Verification, Validation, and Testing (VV&T) are also applicable when specified system requirements are also security requirements. For a general discussion of security evaluation and the four communities in which this activity occurs (i.e., risk analysis, VV&T, security safeguard evaluation, and EDP audit) see FIPS PUB 102, Section 1.5 [FIP102].

4.1 SECURITY EVALUATION METHODOLOGIES

The following methodologies are summarized in this section:

- Touche Ross & Co. [MAI76]
- Peat Marwick Mitchell & Co. [PMM80]
- SRI International/USC Information Sciences Institute (ISI) [NEM78]
- Department of Defense (DoD) [DOD83]

- Testing
- Canadian Institute of Chartered Accountants [CIC75]
- Arthur Andersen & Co. [AAC78]
- AFIPS Security Checklist for Computer Center Self Audits (Peter Browne) [AFI79]
- Internal Controls for Computerized Systems (Jerry FitzGerald) [FIT78]
- Coopers & Lybrand [C&L82] [HAL85]
- Auditing Computer Systems (Faim Technical Library) [PER81]
- Information Security Handbook (Barry Wilkins) [WIL80]
- Department of Health and Human Services [HHS78] [HHS82]
- Department of Agriculture [DOA80] [DOA84]
- GAO Audit Guides [GAO81-1] [GAO81-2]
- Department of Energy [DOE83] [DOE84]
- Formal Verification

4.1.1 Touche Ross & Co. [MAI76]

4.1.1.1 Description -

Who developed it and when?

The Touche Ross & Co. methodology [14] described here is that from William Mair, Donald Wood, and Keagle Davis, Computer Control & Audit, revised in 1976 and published by The Institute of Internal Auditors [MAI76]. It was "developed from actual field experience by practicing internal auditors and certified public accountants" [p. iv]. The methodology has been used inside the firm for eleven or twelve years and outside since late 1973. At least 50-75 thousand copies have been distributed including Spanish versions for Mexico and South America. It has been estimated that the methodology may have "hundreds" of users [DAV80].

---

[14] Since the Touche Ross & Co. methodology is in the open literature and available to anyone who purchases their book [MAI76], a larger space has been devoted to describing their methodology.

What is it?

The book is a "comprehensive manual on computer audit and control [which] outlines within it a methodology for evaluating the process of internal controls in computer systems" [p. iv]. It explicitly points out that it is not a reference manual and does not present a "cookbook" solution [p. 2] to the security evaluation problem.

What are its objectives?

A prime objective of the book is to clarify for auditors "what is meant by adequate control in data processing" [p. iv]. The goal of the compliance audit methodology as defined is "to identify and to verify the existence and effective operation of controls over a specific information processing function" [p. 51]. Another objective is "to predict the reliability and related exposures which should be expected from the system in operation in the future" [p. 20].

What is its scope?

The method addresses all facets of security (e.g. physical, administrative, hardware/software) but is not strongly oriented towards newer technology such as distributed systems. Detailed approaches are suggested for three types of audit areas: applications, the system development process, and the information processing facility.

What communities or environments does it support?

It is tailored for analysis of an EDP department within an organization, where distributed technology is not heavily used.

How does it work?

There are seven major steps in the overall audit process [pp. 52-56]:

- Define objectives
- Obtain a basic understanding of the area being audited
- Obtain a detailed understanding of the area
- Identify and evaluate critical controls, processes, and apparent exposures
- Design the audit procedures (tests)
- Test the critical controls, processes, and apparent exposures
- Evaluate and report on the results of the tests

The first three steps in this process are standard in auditing literature. The methodology being discussed here represents the Touche Ross & Co. approach to the fourth step. This approach is introduced as follows [pp. 181, 182]:

> "[At this point], the auditor customarily studies the detailed information that was gathered and then intuitively leaps to a conclusion. Such intuition is largely dependent upon the skills and experience of the auditor. The quality of the decision becomes quite suspect when the application system is sophisticated or unique.
>
> Review of such conclusions may be approached in two ways. A reviewer may simply accept the judgment of the auditor, or he may restudy the detailed information gathered and reach his own independent conclusion. The first of these approaches requires a substantial amount of faith, and the second requires a substantial amount of time. Often neither approach is suitable or justifiable in the circumstances.
>
> We believe that a third approach is feasible. The process of evaluating controls <u>can</u> be made into a formal, reviewable process supported by documentation. The steps in this process are as follows:
>
> - Segregate and classify controls and activities subject to control.
> - Subjectively quantify the effectiveness of purported controls over the various causes of exposure.
> - Identify one or more key controls that should effectively act upon each of the potential causes of exposure.
> - Identify those causes of exposure over which sufficient controls do not appear to exist.
> - Subjectively quantify the business exposures that would result from an undetected occurrence of causes of exposure that lack adequate controls.
> - Select application features to be tested.
>
> On the basis of this selection, the auditor may devise appropriate auditing procedures."

Table 4-1. Touche Ross Application Activities Subject to Control [MAI76, p. 62]

- o Initiate
- o Record
  - Recording
  - Coding
  - Transcription
- o Transmit
- o Processing
  - Comparison
  - Calculation
  - Updating
  - File Maintenance
  - Summarization
  - Sorting
- o Data Storage
  - File
- o Output Preparation
  - Reporting
  - Working documents
  - Reference documents
- o Inquiry

As noted under the discussion of scope, the methodology includes three specific implementations, i.e., in the areas of applications, systems development, and the information processing facility respectively. The activities subject to control are different within each area. In the applications area, activities include those shown in Table 4-1. Systems development activities include the phases in a development cycle, e.g., system planning, user specifications, technical specifications, and so forth [15]. Major activities in the information processing facility area are data conversion, computer operations, file/program libraries, and output distribution. The computer operations activity includes both those functions performed by the operator such as mounting files, loading programs, and performing maintenance; and those functions performed by the operating systems which include editing of input job accounting, task management, and so forth. Based on this guidance, then, the detailed identification of activities is an important early step in the methodology.

Remembering that the overall purpose of the method is to evaluate

---

[15] See [BIG80] for the Touche Ross & Co. document on "Managing Systems Development Process."

internal controls and that the purpose of controls is the reduction of exposures, another early step in the method is to identify the exposures which controls should prevent, detect, or correct. The list of exposures used throughout the book is shown in Table 4-2. An exposure can be thought of as the effect of a cause (i.e., the damage caused) and may be stated in dollars multiplied by the probable frequency of its occurrence, much like a loss estimate in risk assessment.

Table 4-2. Exposures [MAI76, p. 76]

Erroneous record keeping
Unacceptable accounting
Business interruption
Erroneous management decisions
Fraud
Statutory sanctions
Excessive costs/deficient revenues
Loss or destruction of assets
Competitive disadvantage

The next step is to identify the causes of the exposures (or threats). The causes must exist before exposures result. A sample list of causes of exposures (for applications) is shown across the top of Table 4-3. Tables as shown in 4-3 and 4-4 are then used to examine the relationships between the activities subject to control, causes of exposure, and the exposures themselves. Since various exposures would not normally arise with equal probability, the likelihood of each exposure is estimated by placing numerical values opposite each exposure according to the following key:

3 - Virtually certain
2 - Probable
1 - Possible but unlikely
Blank - Very unlikely

This tabular analysis helps to systematically examine the "threats" and exposures within each activity.

Next, controls are listed, organized, and weighed against causes of exposure (threats) and exposures in a control evaluation table. A sample from the applications area is shown in Table 4-5. A key is included which indicates the relative strength of the controls (i.e. as very reliable, moderately reliable, useful but not reliable, or of no significant use). Factors influencing this evaluation include whether the controls are manual or computerized and whether they're discretionary or non discretionary. Having developed the control evaluation table, the next step is to evaluate the quality of control, analyzing the controls to determine whether they are effectively

implemented. Ineffective or nonexistent controls are deleted from the table. Then each cause of exposure is reviewed for the controls over it. Next a judgment is made regarding the likelihood that each cause of exposure could occur, remain undetected, or fail to be corrected. Finally a judgment is made regarding the probable exposures.

The next step is to identify the key controls. Detective controls tend to be the most important followed by corrective and preventive controls. Having done this, the auditor must then identify those causes of exposure over which sufficient controls do not appear to exist. In preliminary evaluations, a simple rule of thumb is cautiously suggested, i.e., one highly reliable control (number 3 on the control evaluation table) over a particular cause should be adequate. In the absence of a highly reliable control, two or more moderate strength (i.e., 2) controls or a very large number of useful but not especially effective controls (i.e., 1) would be needed. It is noted that "at some point, probably no quantity of weak controls makes any particular difference" [p. 188]. It is important in this process that users understand the dangers of controls. Some are effective; some are efficient; some are both. Some are dangerous in that they look effective but aren't. The nature, quality, and objectives of the controls as well as their interrelations must be kept in mind.

The next step, often omitted in practice, is to quantify the resultant business exposures. This involves estimating the maximum loss in dollars that could result from an exposure and multiplying the estimate by the likelihood of its occurrence.

The final step in the methodology is to select features to be tested. At this point we return to the final stages of the overall audit process. These include designing, performing, evaluating, and reporting on the tests. Based on the results of tests, the auditor would probably return and re-evaluate the controls based upon the new information. This final evaluation would differ from the original preliminary evaluation because the auditor would now know whether the controls were implemented effectively. This iterative cycle may recur several times.

How is it used?

Before the method can be used in a company, it must be tailored to the company and company personnel trained. Touche Ross & Co. offers a one week training program which involves walking the user through a case study. A fairly extensive book of tutorial case studies has been prepared and can be used to support this training [NEN78]. The case studies were prepared by the University of Illinois, under a grant from the Touche Ross Foundation, for use in classroom teaching of EDP auditing.

Table 4-3 [MAI76, p. 75]

RELATIONSHIPS OF ACTIVITIES SUBJECT TO CONTROL TO CAUSES OF EXPOSURE

	CAUSES OF EXPOSURE																								
	INPUT							PROCESS								OUTPUT					OTHER			
ACTIVITIES SUBJECT TO CONTROL	LOST	DUPLICATED	INACCURATE	MISSING DATA	NEVER RECORDED	BLANKET AUTHORIZATION	INITIATE INTERNALLY	WRONG FILE	WRONG RECORD	INCOMPLETE	INCORRECT	UNTIMELY	INAPPROPRIATE	FILE LOST	PROGRAM LOST	PEOPLE LOST	IMPROPER DISTRIBUTION	LATE OR LOST	OBVIOUSLY ERRONEOUS	ERRONEOUS BUT PLAUSIBLE	EXCESSIVE ERRORS	UNSUPPORTABLE RESULTS	SHADOW SYSTEMS	UNLIMITED ACCESS	MANAGEMENT OVERRIDE
Initiate	X	X	X	X	X	X	X									X						X	X	X	X
Record	X	X	X	X			X							X		X					X		X	X	X
Transcribe	X	X	X	X			X							X		X					X		X	X	
Code			X				X							X		X					X		X	X	
Transmit	X	X	X	X			X							X		X					X		X	X	X
Compare								X	X		X	X	X	X	X	X					X		X	X	X
Calculate									X	X	X	X	X	X	X	X					X		X	X	
Update								X	X	X	X	X	X	X	X	X									X
File Maintenance								X	X	X	X	X	X	X	X	X					X		X	X	X
Summarize								X	X	X	X	X	X	X	X	X					X		X	X	
Sort								X			X			X	X	X					X		X	X	
File								X	X	X	X		X	X	X	X	X	X			X		X	X	
Report								X	X		X	X	X	X	X	X	X	X	X	X	X		X	X	X
Working Document								X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	
Reference Document								X	X	X	X		X	X	X	X	X	X	X	X	X		X	X	
Inquiry								X	X					X	X	X					X		X	X	X

Table 4-4 [MAI76, p. 76]

APPLICATION RELATIONSHIPS OF CAUSES AND EXPOSURES

IMPACT OF CAUSES 3 — Very likely to occur 2 — Likely to occur 1 — May occur Blank — Generally little effect	CAUSES OF EXPOSURES																									EXPOSURES
	INPUT							PROCESSING									OUTPUT						OTHER			
	LOST	DUPLICATED	INACCURATE	MISSING DATA	NEVER RECORDED	BLANKET AUTHORIZE	INITIATED INTERNALLY	WRONG FILE	WRONG RECORD	INCOMPLETE	INCORRECT	UNTIMELY	INAPPROPRIATE	FILE LOST	PROGRAM LOST	PEOPLE LOST	IMPROPERLY DISTRIBUTED	LATE OR LOST	ERRONEOUS BUT PLAUSIBLE	OBVIOUSLY ERRONEOUS	EXCESSIVE ERROR CORRECTION	UNSUPPORTABLE	SHADOW SYSTEM	UNLIMITED ACCESS	MANAGEMENT OVERRIDE	
	3	3	3	3	3	2	2	3	3	3	3	3	3	3		2	2	2	3	2	2	1	1	1	1	Erroneous record keeping
						2	2				2	2	3			2			2			2			1	Unacceptable accounting
	1	1	1	1		1				1	1	1	1	2	2	2	2	2	2	1	2	1		2	1	Business interruption
	2	2	2	2	2	1	1	2	1	2	2	2	2	2	2	2	2	2	3	1	2	1	1		1	Erroneous management decisions
		1	1	1	1	1	1		1	1	1		1			1	2	1	1	1	1	2	1	2	1	Fraud
	2		1	1	1	1			1	1	1	1	1	1		1	1	1	1	1	1	3		1	1	Statutory sanctions
	2	2	2	2			1	2	2	2	2	1	2	2	3	3	2	2	2	2	3		3	1	1	Excessive costs/deficient revenues
						1	1	2	2		2		2	2		1	1	1						2	1	Loss or destruction of assets
	1	1	1	1	1				1	1	1	1	1	1		1	1	1	1	1				1	1	Competitive disadvantage

Table 4-5 [MAI76, p. 185]

APPLICATION CONTROL EVALUATION TABLE

PREVENTION CONTROLS	REFERENCE	APPLICATION CAUSES OF EXPOSURES: INPUT — LOST	DUPLICATED	INACCURATE	MISSING DATA	TRANSACTIONS NEVER RECORDED	BLANKET AUTHORIZE	INITIATED INTERNALLY	PROCESSING — WRONG FILE	WRONG RECORD	INCOMPLETE	INCORRECT	UNTIMELY	INAPPROPRIATE	FILE LOST	PROGRAM LOST	PEOPLE LOST	OUTPUT — IMPROPERLY DISTRIBUTED	LATE OR LOST	ERRONEOUS BUT PLAUSIBLE	OBVIOUSLY ERRONEOUS	EXCESSIVE ERROR CORRECTION	UNSUPPORTABLE	OTHER — SHADOW SYSTEM	UNLIMITED ACCESS	MANAGEMENT OVERRIDE
Definition of responsibilities		1	2	2	2	2	2	2	1	1	2	2	2	1	1	1	1	1	2	1	1	1	2	2	2	
Reliability of personnel		1	1	1	1	1			1	1	2	2	2	1	2	1	2	2	1	1	2	2	2	2	2	1
Training		1	1	1	2	2			1	1	2	2	2	2	1	1		2	1	2	2	2	2	2		1
Competence		1	1	1	2	2			2	1	1	2	2	2	1	1	1	2	1	2	2	2	2	1	2	1
Mechanization		2	2			1		3	1	2	2	2	3	2						1	1	2	1		1	2
Segregation of duties									1							1		1		1	1	1		1	2	2
Rotation of duties																									2	
Standardization		1	2	1	2				1	1	2	2	1	1	1	1		2	2		1	1		2		1
Authorization		1	2	1	1	2			1	1	1			2	2	2		1	1	1	2	1	1	1	2	1
Secure custody		2							2						2	2		2	1				1		3	
Dual custody		1							2						2	2		2	1				1		3	
Forms design				2	2	1																	1			
Prenumbered		2	2								2	2														
Preprinted				2	2							2	2									1				
Simultaneous preparation		2										2	2									1				
Turnaround document			3	2	2	3						2	2													
Drum card				2	1																					
Endorsement		2									2	2		2												
Cancellation			2																							
Documentation		2		2		2		2	2			1	1	2	2	2		2	2	1	2	2	3	1		2
Exception input											2															
Default option											2															
Passwords									1	1				2											3	
		3	3	3	3	3	2	2	3	3	3	3	3	3	3		2	2	2	3	2	2	1	1	1	1
							2	2				2	2	3			2			2			2			1
3 — Very likely to occur		1	1	1	1		1				1	1	1	1	2	2	2	2	2	2	1	2	1		2	1
2 — Likely to occur		2	2	2	2	2	1	1	2	1	2	2	2	2	2	2	2	2	2	3	1	2	1	1		1
1 — May occur			1	1	1	1	1	1		1	1	1		1			1	2	1	1	1	1	2	1	2	1
Blank — Generally little effect		2		1	1	1	1			1	1	1	1	1	1		1	1	1	1	1	1	3		1	1
		2	2	2	2			1	2	2	2	2	1	2	2	3	3	2	2	2	2	3		3	1	1
							1	1	2	2		2		2	2		1	1	1						2	1
		1	1	1	1	1				1	1	1	1	1	1	1	1	1	1	1	1	1			1	1

RELIANCE ON CONTROLS
3 — Reliably controls applicable cause
2 — Controls cause but should be accompanied by additional controls
1 — Useful but not especially affective
Blank — No significant contribution

EXPOSURES
- Erroneous record keeping
- Unacceptable accounting
- Business interruption
- Erroneous management decisions
- Fraud
- Statutory sanctions
- Excessive costs/deficient revenues
- Loss or destruction of assets
- Competitive disadvantage

Warning: Reliance and impact relationships must be tailored to individual circumstances.

Page 1 of 3

Table 4-5 (Cont'd) [MAI76, p. 186]

APPLICATION CONTROL EVALUATION TABLE

		APPLICATION CAUSES OF EXPOSURES																										
		INPUT							PROCESSING									OUTPUT						OTHER				
DETECTION CONTROLS	REFERENCE	LOST	DUPLICATED	INACCURATE	MISSING DATA	NEVER RECORDED	BLANKET AUTHORIZE	INITIATED INTERNALLY	WRONG FILE	WRONG RECORD	INCOMPLETE	INCORRECT	UNTIMELY	INAPPROPRIATE	FILE LOST	PROGRAM LOST	PEOPLE LOST	IMPROPERLY DISTRIBUTED	LATE OR LOST	ERRONEOUS BUT PLAUSIBLE	OBVIOUSLY ERRONEOUS	EXCESSIVE ERROR CORRECTION	UNSUPPORTABLE	SHADOW SYSTEM	UNLIMITED ACCESS	MANAGEMENT OVERRIDE
Anticipation		3				3					3		1		2	2	2	3	3							
Transmittal document		2	2	2							2							2								
Batch serial numbers		3	3						2						3											
Control register		2	2	2					2		2				2											
Amount control totals		3	3	2					3		3	2														
Document control count		3	3						2			1														
Line control count		3	3						2		2	1														
Hash totals		3	3	2					3		2	1														
Batch totals		3	3	2					3		2	2														
Batch balancing		3	3	2					3		2	2														
Visual verification			1	1	2				2	1	1	1		1				2			3					
Sequence check		2	2	1		2				2	2	2			1											
Overflow check				2					2	2	2	2		2												
Format check				2					2	2																
Completeness check					2				2	2	2	2	1	2						1	2					
Check digit				3					2																	
Reasonableness		1	2	2	2			2	2	2	2	2	1	2							2					1
Limit check				2				2	1			2									2					
Validity check				2					2																	
Readback		2		3	2									2							2					
Dating			1			2			2				2	2	2								1			
Expiration			2			2			2					2	1											
Keystroke verification				2	2																					
Approval			1	2	2		2		2		1	2	1	2							3		3	2	3	1
Run-to-run totals									3		3	2			3						3					
IMPACT OF CAUSES		3	3	3	3	3	2	2	3	3	3	3	3	3	3		2	2	2	3	2	2	1	1	1	1
							2	2				2	2	3			2			2			2			1
3 — Very likely to occur		1	1	1	1		1				1	1	1	1	2	2	2	2	2	2	1	2	1		2	1
2 — Likely to occur		2	2	2	2	2	1	1	2	1	2	2	2	2	2	2	2	2	2	3	1	2	1	1		1
1 — May occur			1	1	1	1	1	1		1	1	1		1			1	2	1	1	1	1	2	1	2	1
Blank — Generally little affect		2		1	1	1	1			1	1	1	1	1	1		1	1	1	1	1	1	3		1	1
		2	2	2	2			1	2	2	2	2	1	2	2	3	3	2	2	2	2	3		3	1	1
							1	1	2	2		2		2	2		1	1	1						2	1
		1	1	1	1	1				1	1	1	1	1	1	1	1	1	1	1	1	1			1	1

RELIANCE ON CONTROLS
3 — Reliably controls applicable cause
2 — Controls cause but should be accompanied by additional controls
1 — Useful but not especially effective
Blank — No significant contribution

EXPOSURES
Erroneous record keeping
Unacceptable accounting
Business interruption
Erroneous management decisions
Fraud
Statutory sanctions
Excessive costs/deficient revenues
Loss or destruction of assets
Competitive disadvantage

Warning: Reliance and impact relationships must be tailored to individual circumstances.

Page 2 of 3

Table 4-5 (Cont'd) [MAI76, p. 187]

APPLICATION CONTROL EVALUATION TABLE

DETECTION CONTROLS (continued)	REFERENCE	APPLICATION CAUSES OF EXPOSURES: INPUT: LOST	DUPLICATED	INACCURATE	MISSING DATA	NEVER RECORDED	BLANKET AUTHORIZE	INITIATED INTERNALLY	PROCESSING: WRONG FILE	WRONG RECORD	INCOMPLETE	INCORRECT	UNTIMELY	INAPPROPRIATE	FILE LOST	PROGRAM LOST	PEOPLE LOST	OUTPUT: IMPROPERLY DISTRIBUTED	LATE OR LOST	ERRONEOUS BUT PLAUSIBLE	OBVIOUSLY ERRONEOUS	EXCESSIVE ERROR CORRECTION	UNSUPPORTABLE	OTHER: SHADOW SYSTEM	UNLIMITED ACCESS	MANGEMENT OVERRIDE
Balancing		2	2	2	1				3		3	2									3					
Reconciliation		2	2	2	2	2			3	2	3	2		1						2			2			1
Aging		2	2	1	1	2				2	2	2	2	1								2	1			
Suspense file		2	2			2				2	2		2	1								2				
Suspense account		2	2			2					2		2	1									1			
Matching		3	3	3	2				3	3	2	1	2	1									1			
Clearing account		2	2	2	2	2					2		2	1									2			
Tickler file		2				1							2													
Periodic audit		2	2	3	2	2		3	3	2	2	2	2	2	2	2	2	2	2	2	3	2	3	2	3	1
Redundant process				2	2							1		2												
Summary process				2	2		2				3	1		2												
Label		1	1						2						2	2		2							1	
Trailer record		2	2	1	1				3		3				2											
CORRECTION CONTROLS																										
Discrepancy reports		2	2	2	2	2			3	3	3	3		2					2							1
Transaction trail			3	3	2	2				3	2	3	2	2												1
Error source statistics		2	2	2	2					2	2	2		2								3				1
Automated error correction				2	2						2	2		2								3				
Upstream resubmission				2	2					2	2	2		2												
Backup and recovery		3													3	3	3	2	3						2	
		3	3	3	3	3	2	2	3	3	3	3	3	3	3		2	2	2	3	2	2	1	1	1	1
							2	2				2	2	3			2			2			2			1
3 — Very likely to occur		1	1	1	1		1				1	1	1	1	2	2	2	2	2	2	1	2	1		2	1
2 — Likely to occur		2	2	2	2	2	1	1	2	1	2	2	2	2	2	2	2	2	2	3	1	2	1	1		1
1 — May occur			1	1	1	1	1	1		1	1	1		1			1	2	1	1	1	1	2	1	2	1
Blank — Generally little effect		2		1	1	1	1			1	1	1	1	1	1		1	1	1	1	1	1	3		1	1
		2	2	2	2			1	2	2	2	2	1	2	2	3	3	2	2	2	2	3		3	1	1
							1	1	2	2		2		2	2		1	1	1		1				2	1
		1	1	1	1	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1			1	1

RELIANCE ON CONTROLS
3 — Reliably controls applicable cause
2 — Controls cause but should be accompanied by additional controls
1 — Usaful but not especially effective
Blank — No significant contribution

EXPOSURES
Erroneous record keeping
Unacceptable accounting
Business interruption
Erroneous management decisions
Fraud
Statutory sanctions
Excessive costs/deficient revenues
Loss or destruction of assets
Competitive disadvantage

Warning: Reliance and impact relationships must be tailored to individual circumstances.

Page 3 of 3

What skills are needed to successfully use it?

An auditor with ADP skills is needed to carry out the methodology. As the book notes, "Evaluation of control strengths and weaknesses is a highly subjective process. No pat formula or procedures exist to give the answers or even to make it easy. This is the process which most requires professional skills of the auditor" [p. 181].

What inputs or data are needed to exercise it?

System documentation, interviews, organization documentation, and so forth serve as inputs. "The ability to review system documentation at the logic level is probably the most critical requirement in the audit of EDP applications" [p. 50].

4.1.1.2 Distinguishing Features - One distinguishing characteristic of this methodology is the fact that it is presented in "book" form, with the book serving as a lengthy, insightful tutorial addressing the overall audit process. Another distinguishing feature is the structured use of matrices to evaluate application security.

4.1.1.3 Notable Experiences And Lessons - Del Monte prepared a major case study in 1980 on how they have used the method. They built their own evaluation matrices in the process and have incorporated the methodology into their system development method. The study was published by the Institute for Internal Auditors [SMI80].

4.1.1.4 Major Strengths And Weaknesses - The methodology's primary strengths are its thoroughness and structure. It conveys an insightful awareness of the complexity and limitations associated with evaluation. It forces and structures thought, providing a focal point for controlled analysis and documentation. No final "score" is produced. In fact, major misunderstandings have resulted from the limited quantitative ratings that exist (e.g. via the assumption that a high score means an area is more secure than one with a low score). As a result, in the future the authors will be removing the use of numbers entirely and replacing them with letters to help avoid numeric oversimplification.

4.1.2 Peat Marwick Mitchell & Co. [PMM80]

4.1.2.1 Description -

Who developed it and when?

Peat Marwick Mitchell & Co. has developed a Data Processing Security Evaluation Guideline (DPSE, pronounced Dipsey). Its development began in 1973 and the firm has used it for "hundreds" of clients. It is a proprietary methodology. The firm is considering licensing it to users.

What is it?

DPSE is a partially quantitative, comprehensive, systematic methodology for evaluating security in a data processing environment. It consists of an embedded installation profile and a series of questionnaires formatted so as to accommodate a scoring scheme for the answers to each question. These scores are then consolidated into a score for each major security area reviewed.

What are its objectives?

The DPSE "guidelines and program are intended to aid professional data processing audit and consultant representatives in planning, reviewing, and evaluating (in-depth), the security of data processing installations" [p. IB-1]. "Although the review is designed for medium to large-scale data processing environments, the comprehensive topics of data processing security are appropriate to all computer installations" [p. IB-1]. The detailed product resulting from the use of DPSE is used to support security recommendations to an organization.

What is its scope?

The PMM "guidelines serve as the basis for evaluating security and controls for both data processing operations and designated computer applications" [p. IB-1]. The methodology is thus clearly applicable to evaluations of applications, systems, and installations. DPSE does not include privacy considerations, however. While typically used internally, DPSE has also been used by auditors for third party reviews (financial audits). The ten areas of security that are addressed are:

- internal audit
- administrative security
- physical security
- standards
- processing security
- operating system security
- software security
- data base security
- communications security
- applications security

Each of these ten areas has then been broken out into a comprehensive set of four to seven subtopics. The questionnaires (checklists) developed in DPSE address numerous relevant questions in each of the subtopic areas.

What communities or environments does it support?

It is intended for analysis of EDP departments within "large and sophisticated data processing" [p. IA-1] facilities of organizations.

How does it work?

First, a client profile is obtained and a high-level "pre-engagement" review is performed to scope the review that will take place. As stated above, it is meant to be an aid to data processing auditors and consultants. An important early step in the application of the method is to tailor each of the ten areas to the specific characteristics of the client. This high level review will consist of a tour of the corporate data processing facility and meetings with senior management, corporate data processing management, internal security, and internal audit. An evaluation of the status of internal audit then takes place. "The work plan for the comprehensive review will largely be dependent on the evaluation of the internal audit" [p. IC-2]. Based on these findings, major topics to be addressed in the evaluation will be selected and quantitative weighting factors will be assigned to the security areas and their subareas, thus emphasizing where the high risks seem to be located. A user group using the Delphi technique may generate the weighting factors. [This general technique obtains a consensus position from a number of persons knowledgeable in a subject.]

The review techniques to be used are selected from among fifty-nine Information System Analysis Techniques. These techniques include such activities as Application System Value Measurements, Computer Network Analysis, Data Handling Analysis, etc.. Data is collected via interviews, observations, documentation, and testing.

Based on the data collected in the questionnaires, scoring of the applicable security subareas takes place. Using the subarea weighting factors, a security area score is generated, with each security area scoring between 20 and 100. (100 means the area has a very high security status.) Comparisons of security level (or score) are valid among security areas evaluated within an organization. Finally, using the area weighting factors, a composite security level score is generated for the organization. Comparisons of these scores between successive security evaluations of the organization have meaning. Comparison of these scores between organizations, however, has no meaning.

The result of the evaluation is a recommendation for security changes. If additional protections are recommended in an area and the organization cannot afford to implement them, added insurance in that area is a suggested alternative. A plan for continued analysis and

review is also generated since PMM & Co. recognizes that security evaluation is a dynamic activity that requires periodic repetition to be effective.

How is it used?

Peat Marwick Mitchell & Co. usually plays a major role in the first performance of an evaluation for each customer. Afterwards, the customer takes over, using DPSE as its own internal tool. The evaluation should be performed every year.

What skills are needed to successfully use it?

Thorough user experience in data processing and security is required to tailor and weight the security analysis areas. Audit experience is a requirement to integrate the entire evaluation. Specialized technical expertise is needed in the individual security areas. In fact, a major reason for segregating the analysis into ten distinct security areas is that this permits different technical experts to more easily participate in the analysis.

What inputs or data are needed to exercise it?

System documentation, interviews, organization documentation, and testing serve as inputs.

4.1.2.2 Distinguishing Features - The primary distinguishing characteristics of DPSE are that it is:

- o proprietary
- o partially quantitative, producing a score
- o based on a Delphi weighting

4.1.2.3 Notable Experiences And Lessons - This firm has used this method for "hundreds" of clients over a period of years and claims it has been a useful effective tool.

4.1.2.4 Major Strengths And Weaknesses - A major strength is that it provides a detailed product to support security recommendations, making it hard to dismiss them as mere opinion. Another major strength is that it enables the security evaluation to build on the work of previous years. Developers agree, however, that the use of the final quantitative score is "not a strength" since it is based on much judgment and can be misinterpreted. However, they find that clients find this an attractive feature.

4.1.3 SRI International/USC Info. Sciences Institute (ISI) [NEM78]

4.1.3.1 Description -

Who developed it and when?

SRI International has performed much work in the audit and evaluation areas, including such landmark work as the Systems Auditability and Control (SAC) study [IIA77] and the subsequent work, the Relative-Impact Measure (RIM) [NIE80] approach to risk assessment. The security evaluation approach summarized here is different from these. It was documented in a paper by Peter Neumann in the 1978 National Computer Conference Proceedings (i.e. "Computer System Security Evaluation") and draws heavily from work done at ISI by Bisbey, Carlstedt, and Hollingworth. The ISI work terminated in approximately 1976 due to funding cut-backs.

What is it?

First, this is an evaluation approach , not an evaluation methodology. It provides the seminal ideas and structures but provides no explicit guidance in their use. The approach is summarized here because it presents an innovative and potentially useful restructuring of the security evaluation process (for systems and applications).

What are the objectives?

A primary objective of the paper was to use the ISI categorization of protection flaws to evaluate SRI's Hierarchical Design Methodology (HDM) and show how HDM can intrinsically avoid the flaws. Another objective was that with which we are concerned, namely, illustrating the significant potential embodied in the combined preventive and remedial approach to computer system security evaluation.

What is its scope?

The approach is applicable primarily to the technical evaluation of computer systems and applications programs. It is not applicable to physical, administrative, or installation security in general. It is theoretically applicable to hardware evaluation. Operating system evaluation would probably be its strongest forum. It is equally applicable to existing systems or those under development.

What communities or environments does it support?

With respect to operating systems, DoD community needs are most in line with the fairly rigorous security standards intended via the approach. Although the approach might be of some use in evaluating typical existing systems, the paper readily states that "conventional commercial operating systems are ignored here, as they are for the most part intrinsically insecure" [NEM78, p. 1092]. With respect to

application programs, the approach is perhaps applicable to a slightly wider community but is still best suited for those environments requiring a high degree of security. This approach might be increasingly applicable in the future, should more secure systems and applications come into wider use.

How does it work?

Two separate "metrics" are used to evaluate the systems or applications: the protection flaw categorization and methodological considerations. Protection flaw categories and symptoms are shown in Tables 4-6 and 4-7. Methodological considerations are shown in Table 4-8. No claims are made that the categories and considerations are complete or commonly accepted. They do, however, serve as a good starting point.

There is no "methodology" for evaluating against these structures. Presumably experienced security experts would review the documentation in question and talk with developers and users. The paper illustrates sample evaluations on two systems: Multics and UNIX (see Tables 4-9 and 4-10). From these tables this approach is clearly seen to be an informal process, with no attempts made to rate individual areas or systematically integrate them into a single system rating.

How is it used?

No mode of use is discussed. The approach essentially represents an evaluation structure which could be used in different ways. It provides a high-level checklist which could be used by the system/application designer and reviewer.

What skills are needed to successfully use it?

Use of experienced security personnel is needed. The paper notes that "given a computer system that has been expressly designed to be secure, it is still a difficult matter to assess how secure the system really is" [NEM78, p. 1094]. It is probably not realistic to expect any methodology to enable inexperienced people to evaluate the internal security of an operating system or complex application.

What inputs or data are needed to exercise it?

Design and implementation documentation as well as interviews with developers would be required.

Table 4-6.

[NEM78, p. 1090]

Categories of Protection Flaws (and examples) (Based on Bisbey, Carlstedt, Hollingworth at ISI)

1. Incorrect choice of protection domain or security partition
   (a security-critical function manipulating critical data directly accessible to the user; incorrect initial assignment of security or integrity level at system generation, configuration, or initialization)
2. Exposed representations or implementation detail
   (bypassing an abstraction, e.g., direct manipulation of a hidden data structure such as unmediated user modification of a directory entry; user use of an absolute I-O address; note that the visibility of timing information provides a generic leakage channel, e.g., drawing inferences from page fault activity)
3. Inconsistency of data over time
   (noninvariance of parameters, e.g., change in value of a parameter in a call by reference; change in a file accessible to different processes, e.g., in an improperly protected, shared process directory)
4. Naming problems
   (aliasing, e.g., two distinct names for the same object not being treated identically; ambiguity resulting from use of the same local name for two distinct objects [which may also involve a residue problem, e.g., if name persists in some system table])
5. Residues in allocation and deallocation
   (incomplete deletion, revocation, or deallocation, e.g., such that an apparently deleted value is still accessible—in core, disk, archive store, etc.; incomplete cleanup on abort; ignoring terminal hangup).
6. Nonvalidation of critical conditions and operands
   (invalid or unconstrained parameters such as an out-of-bounds virtual address or absolute I-O address; lack of strong type checking, e.g., a pointer to a structure of the wrong type; absence of quota limit stops such as bounds on queue sizes or number of processes, with overflows resulting in possible system or user crashes)
7. Indivisibility problems (in multiprogramming)
   (interrupted atomic operations, e.g., incomplete interrupt handling [quit during login resulting in partial success, or in perpetual lockup of interlocked data]; faulty read-alter-rewrite in hardware)
8. Serialization problems (in multiprogramming, multiprocessing)
   (incorrect sequencing [e.g., wrong order], improper isolation of atomic operations from one another [e.g., reading during writing, or concurrency among different directory commands on the same directory]; critical race conditions in implementation; deadlocks and deadly embraces)
9. Incorrect choice of operation or operand
   (use of the wrong function, producing incorrect results; use of an unfair scheduling algorithm, producing correct results for each scheduled process, but denying service completely to certain users)

Table 4-7.

[NEM78, p. 1090]

Symptoms of Potential Protection Flaws, by Category

1. Domain choice. All programs or human actions relating to the initialization or interpretation of protection information are suspect, e.g., any setting or changing of a security level, particularly any action that lessens security (e.g., downgrading).
2. Exposed representations. Any direct visibility or use of implementation detail is suspect. Any use of absolute addresses for memory or input-output. Nonvirtual resources. Direct access to a data structure that is normally used as an abstract data object. Serial dependence within logically combinational functions.
3. Data inconsistency. A called procedure fetches the value of a parameter more than once, or fetches a value it just stored. A parameter is passed by name or by reference. (The value may change between call and return.) An output value is overlaid on top of an input value. Reference is made to a value that is self-modifying upon being accessed.
4. Naming. Any object for which two different names can exist is suspect. Use of a local name in one context, a global name in another, e.g., a virtual and a nonvirtual name. Any use of a table index where protection is expected.
5. Residues. Physical deletion of contents is suspect whenever deferred beyond logical deletion, e.g., deferred until reuse of media space. Readable free pools. Accessible backup storage. Reuse of an index or slot number after deletion of entry.
6. Nonvalidation. The absence of any checks on protection information upon access to sharable data is usually indicative of a flaw. The absence of any checks on an input variable or parameter, on its type or value range, or even on the existence of data is suspect. Lack of quotas on real resources or on different virtual partitions of resource usage can provide a leakage channel across partitions. Validation of status at the time of a request for which status may change, without revalidation on completion.
7. Indivisibility. Any allegedly noninterruptible or indivisible operation is suspect, as is the mechanism for achieving indivisibility.
8. Serialization. Any overlapping of operations using the same data is suspect, either with different uses of the same operation, or simultaneous uses of different operations on the same data base.
9. Choice of operation or operand. This category is very hard to formalize. Potentially any operation can be improperly chosen. Operations and data types that do not correspond are suspect, as is the use of mismatched type declarations.

Table 4-8 [NEM78, p. 1091]

Factors Influencing Defensiveness in Systems and Applications

- Well-defined and well-understood requirements, established clearly and agreed upon in advance
- Good design (e.g., modularly structured, especially hierarchically, with strict isolation of application programs and system programs, strongly typed operations, unified treatment of storage, input-output [e.g., mapped virtual access])
- Suitable implementation languages (e.g., strong typing, avoidance of aliasing, constrained argument passing [such as use of call by value where data inconsistency may be a problem], hiding of implementation detail and device dependence wherever possible, clean control structures, encapsulation of data types)
- Well-defined and understandable specifications for the system hardware and software
- Structured implementation, reflecting the modularity of the design wherever appropriate, and structured initialization (e.g., hierarchical)
- Systematic handling of exception conditions and quota limits
- Auditing and recovery integrated into system design, e.g., hierarchical
- Careful debugging, testing, verification
- Good management of system development (e.g., respecting these factors)
- Lessening the need for management as a result of simplifications resulting from use of these factors
- Good management of system operation (e.g., rigid adherence to system generation and evolution protocols)
- Nonreliance on secrecy of design and implementation
- Awareness of the user community (e.g., enforcing the use of random pronounceable passwords rather than guessable ones)

If formal verification of the design or its implementation is desired, then the following also contribute, both separately and collectively:

- Formally stated requirements
- Formally specified design, including specifications of modules and their interrelationships (e.g., data representations)
- Formal proofs of correspondence between design specifications and requirements
- Formal axiomatization of the programming language
- Formal proofs of consistency of programs with design specifications
- Formal axiomatization of the hardware/microcode
- Formal proofs of consistency of hardware/microcode with hardware specifications

Table 4-9 [NEM78, p. 1093]

Evaluation of Multics and UNIX with Respect to Characteristic Flaws

Category of Flaw	Multics	UNIX
1. Domain choice	Good. System, users in multiple rings. Ring 0 capture omnipotent, but unlikely. Outward migration of less critical functions. Provable layered security kernel designed by Honeywell, but not implemented.	Poor. Nonstratified. Capture of the entire system easy through superuser infiltration. Security kernel implementations exist (UCLA, MITRE), improving on original decomposition.
2. Exposed representation	Access within a ring all or nothing. Considerable hiding via ring mechanism.	Superuser easy to capture, defaults open. Trojan horses galore. Process temporaries readable and writable, main memory readable.
3. Data inconsistency	Argument pointer is copied onto stack by call. Arguments are copied within the system code to avoid this flaw internally.	Argument itself is copied, but only by convention. Process-temporary files are alterable in midcomputation by user or other users.
4. Naming	Collisions of local names: Trojan horse!	Default on directory entries: unprotected. Memory aliases (''/dev/mem''). Trojan horses abound.
5. Residues	Core zeroed only before reallocation, but is not virtually addressable after deallocation. Disk never zeroed, just overwritten. Residues after crash.	Core zeroed only before reallocation, and is until then still readable. Disk never zeroed, just overwritten. Residues after crash.
6. Nonvalidation	Hardware checking at ring and segment levels. All kernel args validated. Compile-time data-type checking.	Very few checks on resource or I-O bounds, interuser ops. Easy to crash.
7. Indivisibility	Locking makes kernel ops atomic. Exception handlers force no-effect noncompletion.	Exception handling bad.
8. Serialization	Locks prevent overlap. Lock hierarchy used to avoid deadlocks.	OK? System functions logically synchronous (until IPC installed).
9. Op choice	Possible problems?	Possible problems?

Table 4-10.
[NEM78, p. 1094]

Evaluation of Multics and UNIX with Respect to Methodological Considerations

Influence	Multics	UNIX
Requirements	Informal	Informal
Specifications	Implementation laden	Implementation laden
Design structure	Implicit hierarchy	Procedural structure
Design	Fairly unified	Fairly clean
Exceptions,quotas	Fairly systematic	Poor
Implementation	Segmentation useful	Device independence OK
Programming language	PL/1 somewhat more helpful than C	C poor on abstraction, strong typing
Verifiability	Difficult because of size, PL/1, no specs	Pointless because of insecurity, C, no specs

4.1.3.2 Distinguishing Features - The primary distinguishing feature of this approach is the protection flaw categorization. Another distinguishing feature is the explicit use of development methodology considerations in evaluating an existing system. In a sense, this factors in the issue of "assurance" which can be thought of as a general quality measure associated with each control in addition to the simple fact of whether the control exists. Justification for this type of approach is in fact stated in another SRI paper by Nielsen and Ruder (which describes the relative impact measure (RIM) approach to risk assessment) [NIE80, p. 21]: "Since it is very difficult to determine the existence of exploitable system flaws in a given computing facility, the methodology instead uses the concept of system susceptibilities to various types of flaws." Consideration of development methodology allows inferences to be made about the "likelihood-of-existence" of flaws without mounting a penetration or testing effort to actually find them.

4.1.3.3 Notable Experiences And Lessons - There is little documented experience in the use of this approach.

4.1.3.4 Major Strengths And Weaknesses - The major strengths of this approach are its introduction of the ISI protection flaw categorization and its explicit incorporation of methodological considerations into an evaluation process. The major weakness is the total lack of guidance on how to implement this potentially useful structure.

4.1.4 Department Of Defense (DoD) [DOD83]

4.1.4.1 Description -

Who developed it and when?

Security evaluation in the DoD is embedded in the certification process. Certification policies and procedures within DoD have tended to be system or installation specific. The few certification policies which exist are primarily testing based with recent increasing emphasis on design review [NEU80].

One DoD initiative has, however, made promising progress in improving the situation. The primary thrust for this new work was initiated in a session at the NBS/GAO Invitational Workshop in November 1978 on "Audit and Evaluation of Computer Security II: System Vulnerabilities and Controls" [RUT80, pp. 8-1 through 8-28]. The session addressed processors, operating systems, and nearby peripherals and was chaired by Theodore Lee. The work was extended upon by MITRE [NIB79], and sponsored by the DoD Command, Control,

Communication and Intelligence (C3I) Directorate and within its DoD Computer Security Initiative.

The need which gave rise to the overall initiative (of which the evaluation work is a part) can be summarized by several quotations. The first is from Dr. Gerald Dinneen, then Assistant Secretary of Defense (C3I) [DIN79, p. A-3].

> "Building computer hardware and software systems is a very complex process that the Government is no longer directly involved in except for special purpose systems that are unique to our needs. The large majority of our computer systems are purchased from the commercial marketplace. We realize that, if we are to achieve widespread availability of trusted systems, they must come from this same source. The DoD cannot afford, just for the sake of having trusted computer software systems, to develop its own general purpose hardware and software systems."

Expanding on this is a quote from Stephen Walker (C3I), who, at that time, directly oversaw the DoD Security Technical Consortium [WAL80, p. 60].

> "One way to overcome this impasse is for someone (like the DoD) to build a trusted system, demonstrate that it is acceptable in real applications, and provide detailed information on the techniques used in the development to the computer industry. If the technology used to build the system was suitable for application in general sensitive information handling environments, then there is a large and rapidly growing marketplace for such a product."

The major portion of this DoD Security Initiative, then, was to support the development of several "trusted" systems serving as "existence proofs". As another part of this overall program, DoD was to attempt "to establish an efficient and consistent evaluation process for determining the integrity of computer systems and the environments for which a particular system will be suitable" [DIN79, p. A-4]. This evaluation process was expected to take the form of a "Laboratory Evaluation" of industry-developed systems, resulting in an "Evaluated Products List" [WAL79, p. K-4].

In early 1981, DoD requested the National Security Agency to establish a Computer Security Evaluation Center to assist in this effort. The DoD Computer Security Center was created in January 1981 and produced a hierarchical set of security requirements for increasingly secure trusted systems [DOD83]. Within this Center, the Laboratory Evaluation group was formed and has tested a number of products to establish where in the Center's security hierarchy these products sit. As of now (April 1985) one operating system (SCOMP) satisfies the highest level of the security hierarchy; and three add-on packages have been given ratings (RACF,ACF2,and TOP SECRET). It should be emphasized that the process is seen as still in the

developmental stage, subject to further refinement and improvement.

What is it?

This "Laboratory Evaluation" process is a reasonable attempt to structure and implement an evaluation process. It is not a methodology, but instead a fairly systematic approach to the evaluation problem which still relies heavily on qualitative expert judgment. Within the past year a small Evaluated Products List has been generated. DoD has also done work in correlating the trusted system hierarchy with the various DoD environments.

What are its objectives?

Two major objectives are [WAL79, p. K-3]:

- o to establish a consistent evaluation process applicable to systems DoD-wide
- o To avoid multiple evaluations of the same systems for the same application

The pre-1981 DoD security certification-accreditation process is summarized in Figure 4-1. The figure shows that a Designated Approving Authority determines approval for each individual installation based on DoD security policy (i.e. Directive 5200.28 [DOD78]) and installation-specific requirements. Approval derives from evaluations performed covering all facets of security such as physical, administrative, personnel, and computer security. One problem with this is that the Designated Approving Authority differs for different installations and environments. This tends to result in an inconsistent, inefficient approval process. The post-1981 certification-accreditation process is shown in Figure 4-2. For computer security, it includes a two-phase approval process including both laboratory approval and site approval. Laboratory approval involves the centralized evaluation of the design and implementation of industry developed systems resulting in an Evaluated Products List. Site approval would be performed by the Designated Approving Authority according to site-specific requirements as related to "laboratory approval".

What is its scope?

The process is basically oriented around generalized hardware and operating system software. Most of it could also be applied in the application evaluation area although this is not being done by DoD on a centralized basis. The process is not designed for installation security evaluation or such concerns as physical and administrative security.

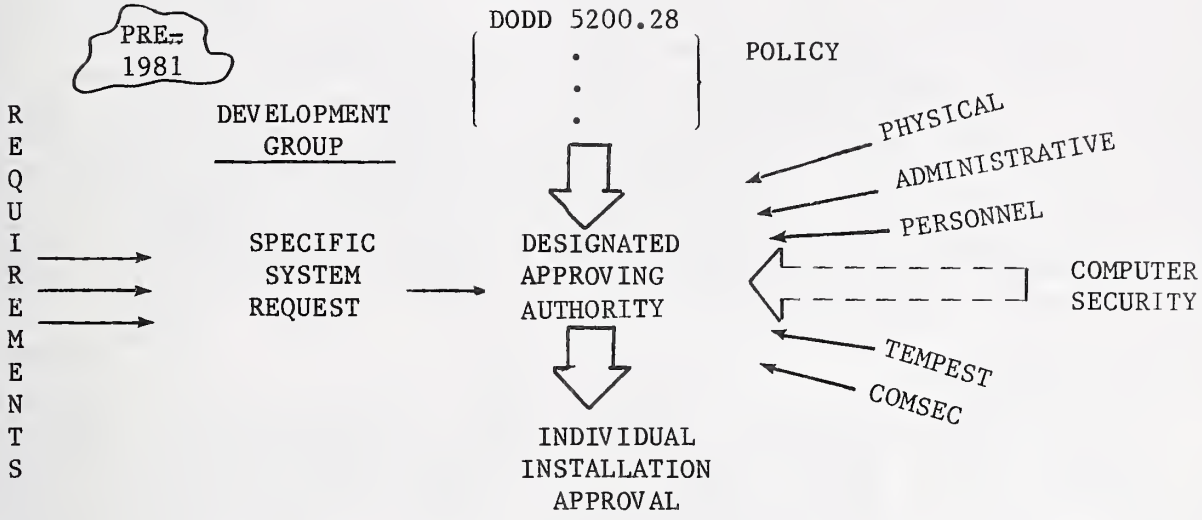


Figure 4-1. PRE-1981 Approval for DoD Use
[WAL79, p. K-2]

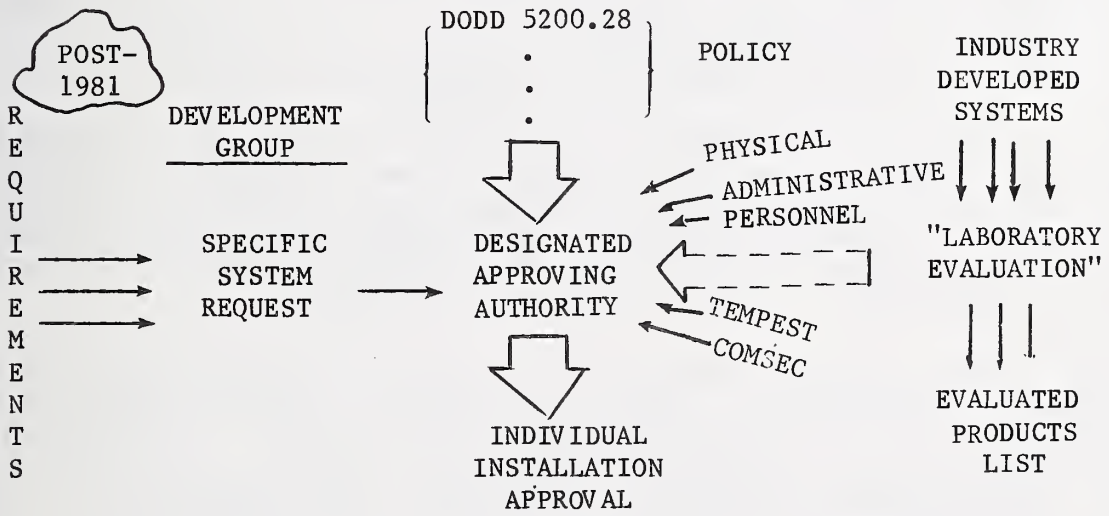


Figure 4-2. POST-1981 Approval for DoD Use
[WAL79, p. K-4]

What communities or environments does it support?

The process is currently tailored for DoD use. The basic evaluation approach (of formulating a centrally evaluated product list) is applicable to any environment but use in other environments would require adoption of new technical evaluation criteria and this would be a much more difficult task than in the DoD case. For example, a working session on a Secure-System Evaluation at the 1979 Summer Study on Air Force Computer Security [AF79] reduced the original seven levels of security protection proposed by MITRE to six, due to "difficulties in finding a clear distinction between the first two levels" which essentially include all existing commercial systems. Similar difficulties would undoubtedly plague attempts to clearly categorize existing systems.

How does it work?

The initial work [RUT80] identified a security metric (see Figure 4-3) comprised of eight security attributes (four assurance features and four protection mechanisms). Within each attribute, the security features were nominally quantified to roughly indicate the contribution to security presented by presence of the features. From this metric, the group drafted a list of six major categories into which DoD systems fall, with the categories characterized by features from the evaluation metric (see Table 4-11). The list included four components:

- Category number
- Allowed kind of applications
- Allowed mix of classification levels
- Major required security metric features

To illustrate, in category 4, the allowed kind of applications are those with no user programming. The allowed mix of classification levels is Top Secret, Secret, and Confidential. Lastly, the major required security metric features are formal design specifications (from the Software area) and system integrity measures (from the Prevention area).

MITRE's work [NIB79] expanded and modified this initial work. It eliminated the high-level distinction between hardware and software attributes and introduced the attributes of design, implementation, and verification. It also introduced recovery as a protection mechanism attribute and removed high-level consideration of human interface concerns and the granularity of protected objects. These latter issues were said to be factors of functionality rather than protection, and as such were said to be of concern within a security level in evaluating suitability for an application. The MITRE protection factors are shown in Figure 4-4. (O & M stands for operation and maintenance.)

Figure 4-3. Security Metric[1] [RUT80, p. 8-16]

[1]The numbers illustrate approximate "contribution to security" made by each feature, with increasing numbers meaning better security. The numbers have no relation to the levels in table 4-11.

Table 4-11. [RUT80, p. 8-24]

Secure System Categories

1. Dedicated Mode
   (any single level)

   Data Security

2. Benign, Need-to-Know Environments
   (any single level)

   Functional Specification
   Reasonable Penetration Results

3. USAF Data Services Center
   (TS-S)

   Reasonable Modern Programming Techniques
   Limited System Integrity Measures

4. No User Programming
   (TS-S-C)

   Formal Design Specifications
   System Integrity Measures

5. Limited User Programming
   (TS-S-C)

   Proven Design Specifications
   Verifiable Implementation
   Limited Covert Path Provisions

6. Full User Programming
   (TS-S-C-U)

   Verified Design
   Automated Test Generation
   Extended Covert Path Provisions
   Reasonable Denial of Service Provisions

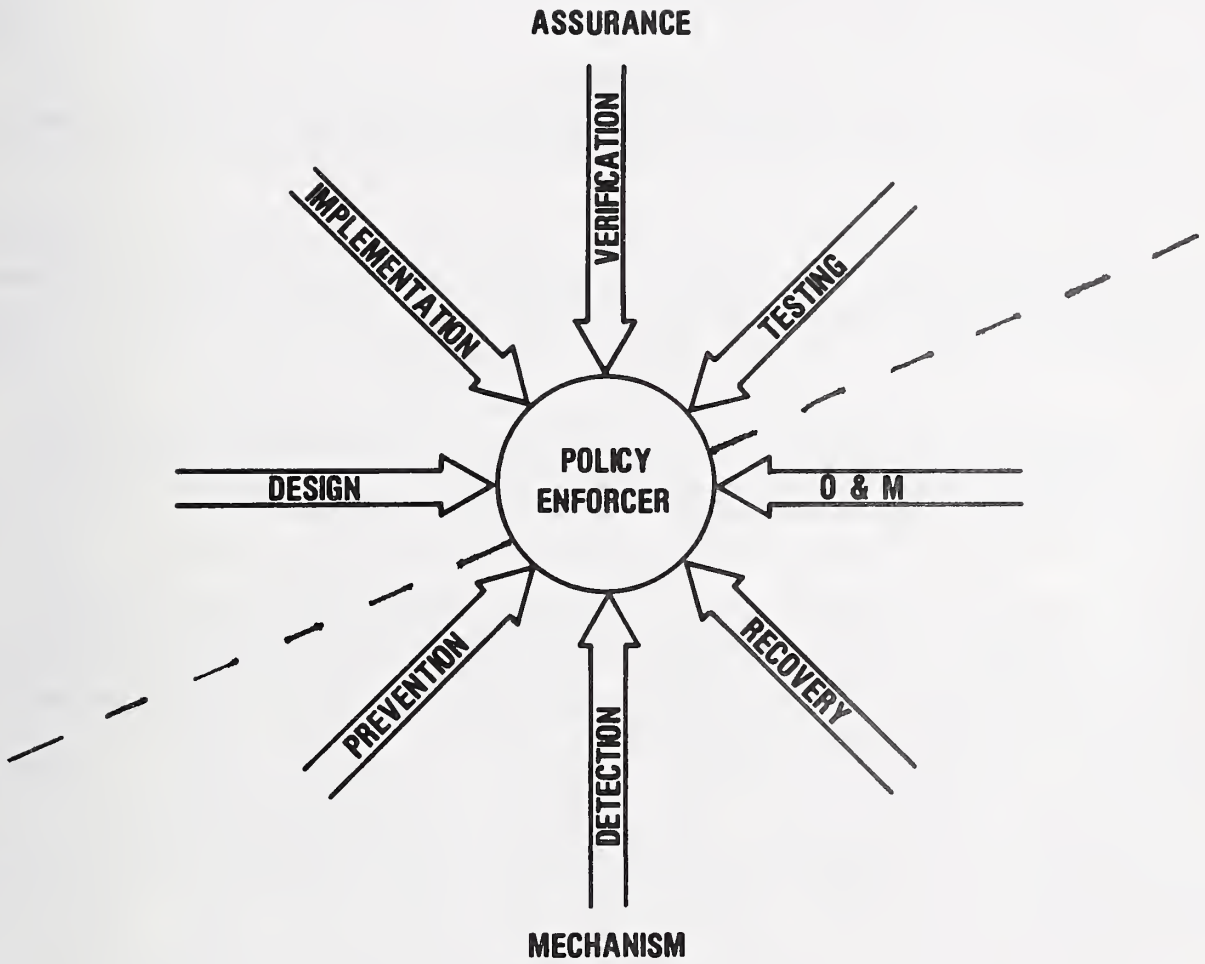


Figure 4-4. Attributes of Trusted Operating Systems [NIB79, p. 4]

The MITRE work went on to configure the evaluation factors into seven levels which are shown in Table 4-12. The primary factors which define the levels are the general security intentions of the developers and the extent of use of formal development methods. MITRE gives assurance features far more weight than the presence or absence of particular protection mechanisms.

Table 4-12. Protection Levels [NIB79, p. 18]

o At level 0 (No Protection), there is no basis for confidence in the system's ability to protect information.
o At level 1 (Limited Controlled Sharing), recognition of some attempt to control access is given, but only limited confidence in the viability of the controls is indicated.
o At level 2 (Extensive Mandatory Security), minimal requirements on the protection policy must be satisfied; assurance is derived primarily from attention to protection during system design and extensive testing.
o At level 3 (Structured Protection Mechanism), additional confidence is gained through methodical construction of the protection-related software components of the operating system (i.e., the Trusted Computing Base (TCB) implementation), and modern programming techniques.
o At level 4 (Design Correspondence), formal methods are employed to verify the design of the TCB implementation.
o At level 5 (Implementation Correspondence), formal methods are employed to verify the software implementation of the design.
o At level 6 (Object Code Analysis), object code is analyzed and the hardware support is strengthened.

Further work expanding on the use of this framework was done at the 1979 Summer Study on Air Force Computer Security [AF79]. The summer study session noted that two application characteristics - user capability and user/data exposure - were of prime importance in determining the level of protection needed. User capability basically refers to the functionality available at the user interface, with the three major levels of capability being function buttons (the simplest interface), transactions, and user programming. User/data exposure refers to the risk deriving from the different classification levels supported. Three major risk levels were identified as follows:

Low (e.g. all data at same level; TS and S)
Medium (e.g. TS/Compartmented and TS; S and U)
High (e.g. TS/C and S; TS and U)

The group used these two application characteristics - user capability and user/data exposure (risk) - to define application classes, and then determined which of MITRE's protection levels were needed for each class (see Figure 4-5). This parallels the sensitivity issue in Federal civilian systems.

More work remains to be done in refining and applying these evaluation criteria. A reasonable consensus exists that this is a good start. More experience is required in the use of these metrics, however. It may turn out that more detailed metrics are not feasible.

This work represents part of the basis for a "Laboratory Evaluation", in which a team of security experts evaluates candidate systems for their security and secureability. The DoD trusted computer system evaluation criteria [DOD83] are providing the additional basis needed for the "Laboaratory Evaluation" of trusted products. There is no detailed methodology for this evaluation process, although a general structure has been defined [TRO80, pp. 11-17]. The structure relies heavily on system manufacturers to make presentations and provide adequate documentation. The actual evaluation is based on the judgment of the evaluators and interaction among them. The final result is the new DoD Evaluated Products List.

How is it used?

The Evaluated Products List represents a laboratory evaluation of the security of a computer system. This will be included as an input to the certification process in evaluating the security acceptability of a specific application of the system in a particular environment as shown in Figure 4-2.

What skills are needed to successfully use it?

Security experts are needed to perform the laboratory evaluation. Little computer security expertise is needed to apply the evaluated products list findings in approving a system for use in a specific application.

CAPABILITY	RISK	PROTECTION LEVEL		SIGNIFICANT ATTRIBUTES
		OS	APPLICATION	
FUNCTION BUTTONS	L	2*-3	3	PROTECTION BASIS IN OS
	M	3	4	CLEAR DEFINITION OF APPLICATION
	H	3*-4	4*-5	
TRANSACTION	L	3-4*	3-4*	4 VS 3 DETERMINED BY METHODOLOGY AND NATURE OF APPLICATION
	M	3-4*	3-4*	
	H	4*-5	4-5	CONFIDENCE OF APPLICATION PROOF RELIES ON PROPER EXECUTION OF OS
PROGRAMMING	L	3-4		CONTROL OVER ACCEPTANCE OF SYSTEM APPLICATIONS AND UTILITIES
	M	5		
	H	6		

* INDICATES MAJORITY FEELING OVER THE RANGE SPECIFIED.

Figure 4-5. Protection-Level User Chart with Added Limitations [AF79, p. 86]

What inputs or data are needed to exercise it?

System documentation, manufacturer presentations, and interviews serve as the primary input to the laboratory evaluation process. In using the evaluated products list, an important input would be a listing of changes to the system which had been made since the laboratory evaluation took place.

4.1.4.2 Distinguishing Features - The primary distinguishing feature is the formulation of an Evaluated Products List based on laboratory evaluations. Another distinguishing feature is the establishment of protection levels.

4.1.4.3 Notable Experiences And Lessons - The approach has been used to evaluate several systems. Establishment of the evaluation center at the National Security Agency is still relatively new. Further data is not yet available.

4.1.4.4 Major Strengths And Weaknesses - Both the protection levels and the concept of an evaluated products list seem to be strengths. The laboratory evaluation approach should result in consistent evaluations and reduce redundancy of effort. The approach may also have the added benefit of improving security consciousness among commercial vendors.

There are several potential weaknesses to the approach. Without a specific methodology for the evaluation itself, there may be no record for the detailed evaluation process to review or build upon (in a subsequent evaluation, should major changes occur). Also, since systems typically are being continuously changed, the evaluation process has a moving target. Baselines would have to be defined for evaluated systems with system changes being evaluated as they occur. Another potential weakness would be possible security breaches due to the centralized collection of both proprietary design information and highly sensitive vulnerability information. In a sense, this seems to contradict the basic principle of separation of duties which is founded on the belief that absolute trust should not be placed in any one individual.

Although the defined hierarchical security levels have been listed as a strength, limitations in the definition may also be seen as weaknesses. For example, it is more oriented towards future systems than those available today. Indeed, a possible change to the levels would in fact be the expansion of the lower levels to represent a wider spectrum of existing systems. As another example, the levels are explicitly oriented towards DoD security policy, and would have to be changed to accommodate other policies.

Overall, the strengths are felt to outweigh the weaknesses. Further experience with and expansion of the approach is required, however, before it can be adequately evaluated.

4.1.5 Testing

Testing[16] has long been the primary method of determining the actual level of computer security in an application or system (and sometimes in an installation, although this will be of less concern here). Although testing research holds much promise for the future, techniques have not changed significantly over the years. The primary difference in testing as an evaluation tool over the next decade will probably be in its increased use to support, rather than replace, new evaluation methodologies.

In general, testing can be seen as falling into two categories - external and internal. External testing is also known as "black box" testing. "Acceptance" testing on delivery of a system typically falls into this category. It does not require manipulation or detailed knowledge of the system or application's internal structure. It is concerned primarily with external functionality (typically at the user interface) as well as overall performance and throughput characteristics.

Internal testing is also known as "white box" testing and does require manipulation and detailed knowledge of internal functions. The major example is program testing. Another example is integration testing. From a security perspective, penetration testing can be either internal or external.

Audit "testing" techniques fall into both categories. Snapshot and tracing techniques, for example, are internal. Test Decks and Base Case System Evaluation are external. Other audit techniques such as Integrated Test Facility and Parallel Simulation are philosophically external but require internal modifications. William Perry has essentially expanded on this same observation in noting that the internal testing audit tools are "primarily data processing debugging practices" with other techniques generally being "sophisticated ways of 'auditing around the computer'" [PER77, p. 9].

Despite the important role played by testing in security evaluation, little has been written about security testing in general. The facet of security testing which has received the most coverage is penetration testing. There exist a number of good penetration

---

[16]This section is structured around a general discussion of testing rather than being oriented around any single methodology. This seems appropriate since there does not exist a single "methodology" of testing - there are many forms. See [POW82] for specific software VV&T tools and techniques.

overviews [LAC74, LID75] and approaches [HOL74, WEB76, WEI73]. A fairly substantial amount of information also exists on penetration findings, although most of this is proprietary or classified. Little visible penetration work is being performed today, however. There are several reasons:

- Early penetration work was founded on an awakened computer security awareness. It demonstrated that essentially any system could be penetrated, even if it were designed with security in mind (e.g., Multics). Once this fact was established, however, the need for penetration efforts decreased, since the vulnerability of a system to penetration was assumed.

- Penetration findings represent bad news. Successful penetrations reaffirm that the system is vulnerable. Yet since systems are assumed to be vulnerable, the inability of a penetrator to subvert a system tends to be seen as a failure of the penetrator, not a success of the system. So neither positive nor negative findings represent "good" news. (Good news might be a new design approach which solves, not reaffirms, the penetration problem.)

- Good penetrators typically make good designers and would prefer to design systems because it is more creative and rewarding and can result in marketable products.

- Most of the work which is being done is not publicized. Organizations are not anxious to advertize shortcomings of their system.

It is true in general that security testing as exemplified by penetration testing has lost its visibility. Current attention instead is often focused on verification and proof-of-correctness techniques which hold the ultimate potential to provide a much better demonstration of security acceptability. Until such techniques arrive, however, testing still represents the best existing way to find the flaws we can comfortably assume to be present. The remainder of this section will summarize some of the testing approaches being used or researched and will identify some of the key security issues associated with testing.

4.1.5.1 External Testing - Perhaps because they are viewed as such mundane topics, very little has been written about both external testing in general and external security testing in particular. Experience, however, shows that there are two aspects of external security testing which are often overlooked. These can be stated as rules:

- o Security "penetration" tests must be independently defined.
- o Precise pass/fail criteria must be defined.

4.1.5.1.1 Independent Definition - If rigorous security testing is desired as a facet of external testing, a separate independent set of "penetration" tests should be devised by an independent group of experts who specialize in security concerns for the system or application type involved. The "Ware Report" emphasizes this point, saying that a "certification procedure must include a phase that deliberately attempts to penetrate our best designs, and that is conducted by technically competent individuals not part of the design group or of the operating agency, and not administratively responsible to either" [WAR79, p. 44]. The primary theme of these tests would be the exercise of malicious or anomalous actions often during unusual or stress system states. It seems that the thought process required to generate these types of tests is different from that required to demonstrate proper functional operation or even proper handling of common error types.

As noted, this penetration testing should be separate from the primary external testing effort. The team defining the primary external tests should prepare their tests as though no penetration testing were being performed. That is, they should address such issues as proper operation of access control, authentication, and audit mechanisms for both valid and invalid data. This prevents concern about scope of responsibility (since the normal external test team is thus responsible for testing the proper functionality of all security controls) and frees the penetration team to concentrate on obscure attacks.

4.1.5.1.2 Pass/Fail Criteria - Most security test plans are quite deficient in defining precise pass/fail criteria for the tests. In part, this derives from ambiguous security requirements. It also derives from incomplete anticipation of likely test outcomes. The issue of rigorous definition of security requirements is a difficult one. It is discussed at the generic level in Chapter 5. Its implications for testing can be illustrated with an example.

Most system requirements definitions include some fairly precise form of throughput or response time requirements. It is usually explicitly stated that this requirement must be met during busy-hour or peak-period operation. What is rarely (if ever) stated is whether this requirement also applies in cases of malicious user attempts to subvert throughput or response time (for others). This is quite a different matter and may require different controls. Requirements would have to state how much degradation in service is acceptable under these conditions in order that the acceptability of test results can be determined.

The second major reason for deficient security test plans is incomplete anticipation of likely test outcomes. Here there is no reason for pass/fail criteria to be ambiguous. Conditions such as the following should be expected to occur during testing:

- The response may be functionally wrong even though there are no negative security impacts.
- The response may be precisely correct, yet slower than the busy-hour response requirement.
- Results may include only a subset of the results expected. On the other hand, the results may include the full set of expected results along with additional extraneous results. Combinations are also possible.
- Tests will intermittently pass and fail.
- Portions of a system may crash without the entire system crashing. A user or group of users may "lock up".
- Situations described ambiguously in design or user documentation (precisely those emphasized during security tests) may result in responses which cannot be precisely predicted.
- Stress testing of one program may cause another module to fail.

All of these can be readily accommodated with precise pass/fail criteria.

4.1.5.2 Internal Testing - The primary form of internal testing is program testing. Miller divides the technology associated with program testing into several categories [MIL78-1, pp. 10-11]:

> Static analysis seeks to demonstrate the truth of certain allegations about program properties without necessarily having to execute the programs.
>
> Dynamic analysis seeks to understand the internal relationships between a program test and the parts of a program that are activated (exercised) during the test.
>
> Test case design attempts to figure out how to construct and/or organize tests to get the best testing effect (highest likelihood of discovering errors) with the least effort.

Symbolic evaluation attempts to determine properties of programs (with a quality level quite close to proof-of-correctness) without actually executing them.

Automated tools provide the technical means to set up, measure, record, and archive the results of testing.

Effectiveness measurement attempts to provide feedback to a user on the quality of past and current testing activities on a particular software system.

Each of these technologies has advantages and disadvantages. Research supports the intuitive notion that no one program testing strategy should be used to the exclusion of others, since different strategies tend to differ substantially in their ability to detect various classes of errors [HOW77, p. 446].

Testing research (e.g., graph theory, cause/effect graphing, reliability theory, etc.) holds promise but many needs remain. Miller lists many including the following from the area of formal testing theory [MIL78-2, p. 10]:

Need-5 : A general theory of formal testing that states, for any given program path, whether or not a particular piece of test data "protects" all of the program text along with that path from all kinds of errors (i.e. both logic and computation).
Need-6 : A method for advising a programmer of the minimum set of additional constraints that must be met by a set of test data in order that the test data reliably be proof against all forms of failure.
Need-7 : A general method for designing the criteria that individual test cases must meet to assure all errors in a program have been protected against. (These criteria would apply before selecting the test data, and could be incorporated in the program text automatically by an automated tool.)
Need-8 : A general method for constructing subsets of test data that meet subset-requirements like those mentioned above for parts of programs.

Improvements in programming and testing methodologies which enhance security testing have been made. For example, in the programming area, modularization and structured programming are being used to produce more reliably testable programs. In the testing area, the ASSERT statement (e.g. Euclid, Ada) is proving useful. The ASSERT statement has been called "the main [testing] invention of the past five years" [MIL78-2, p. 5]. ASSERT statements are included in programs typically as commentary but are sensed during the formal debugging/testing process.[17] "Assertion checker" tools have been developed which can perform this task even when the ASSERT statement

is not available in the language being used. Other software tools have also been developed for security evaluation purposes and actually used in security certification. These include execution path flow analysis and structural analysis tools [NEU80]. In another case, a command interpreter was developed to permit internal system calls (actually security kernel calls) to be made from a keyboard to facilitate internal testing [NAG80]. All of these developments are significant for security. In general, however, improvements in internal testing methodologies have been mostly theoretical.

Two internal testing areas of particular security evaluation interest, though still primarily theoretical, are measures of coverage and software quality metrics.

4.1.5.2.1 Measures Of Coverage - Most program testing approaches are based primarily on the structural analysis of programs. Since this process breaks programs down into a finite number of components, measures of test coverage can be defined based on what percentage of components are exercised by the tests and in which ways. Miller describes such measures [MIL77, p. 201].

> "The most common measure is the C1 measure, which requires that every segment in a program be exercised at least once. [A segment, or predicate outcome, is a program unit with respect to testing. A 1000 statement program typically has about 500 segments.] The C1 measure corresponds with the notion that a program is not well tested unless every decisional outcome has been exercised at least once. A slightly stronger version is the C1p measure, which requires, in addition to the C1 level of coverage, that every predicate term be evaluated at least once to each possible truth value. For example, if the program predicate were A.AND.B the C1p measure would require that both A and B be taken to the true and false outcomes (by comparison C1 requires only that the whole predicate A.AND.B be taken to the true and false outcomes).
>
> At the next level of sophistication there is a class of coverage measures that relate to the checks made of the iterations within a program. The Ck measure requires that every iteration be exercised up to and including k repetitions of the loop; typically, k is set to 2. Ck implies C1.
>
> Another measure of practical interest is the Cd

---

[17] In one application, macro instructions were developed to support the use of both entrance assumptions and exit assertions as an aid in ensuring program integrity [MAN80].

measure, which requires that each dependent pair of program segments be executed together in at least one test. This measure is stronger than C1; provided that the means are available to find the dependent pairs, Cd can be quite an effective determinant of program quality."

For want of better measures, then, one can envision a certification condition requiring key security software to have, say, 98 per cent coverage with a C2 measure while other software may have 95 percent C1 coverage.[18] Of course, the capability should exist to determine which parts of the program are keeping the value from attaining 98 (or 100) percent.

At the integration (i.e. system) testing level, similar coverage measures could be available. Program segments would be replaced by modules. For security evaluation purposes, in configuring programs into modules, it would be preferable to segregate security functions from functions not related to security. This would allow separate coverage measures for security and non-security software and would permit internal integration testing to more readily focus on the system or application's internal security boundaries.

In sum, for internal program or integration testing, coverage measures represent a promising way to quantitatively measure the level of security in a system or application (in terms of freedom from errors). Of course, it should be stressed that even 100 per cent coverage would not defend against security relevant design errors. Nor would it defend against many subtle exploitable errors which might be too complex to be represented by the coverage measure used or might exploit an asynchrony or resource limitation only possible under system conditions not representable in a coverage measure.

4.1.5.2.2 Software Quality Metrics - For several years research has been underway on approaches to measure or predict software quality. Many classes of software quality metrics have been proposed: dynamic, static, quantitative, qualitative, value, check list, pairs, and n-tuple [BOW78, p. 148]. Although these are typically claimed as validated by their developers, there seems to be a consensus that "few of the metrics have been either proven or disproven" [CAV78, p. 135]. The most popular criterion for validation has been number of errors.

Recognizing the many potentially measurable factors of quality, one study proposed a classification of these from the user point of view [CAV78]. Figure 4-6 shows these user-oriented quality factors.

---

[18] The Department of Housing and Urban Development (HUD) has actually implemented simple coverage criteria for certification [SOR79]. Its minimum criteria are "(1) test every branch ('IF' statement) of the program, and (2) test every statement at least once." Certification testing is required of all programs.

In general, there are many factors involved in software quality and many candidate ways to quantify software differences (according to at least some of the factors).

Some of the metrics evaluate software during its development. In this regard, they are not test-like. The set of attributes, or criteria, established for each quality factor then provides a measure to indicate progression towards a desired level of quality.

The overall value of software quality metrics is that they can serve as a supplemental assurance mechanism (along with testing) in evaluating software. Since software quality contributes to the degree of security (especially with regard to vulnerabilities arising from implementation errors), software quality metrics can play a role in evaluating security. Of course the main factors determining the security of a system or application are the high-level factors of policy and design. Software quality metrics tend to be concerned primarily with the implementation. To the extent implementation quality considerations influence security, software quality metrics will be relevant to the security evaluation process. The degree of this relevance will depend upon the situation and the nature and quality of the metrics involved. While the full potential security benefits of these metrics remain to be explored, it can be stated that security evaluation personnel should at least monitor the major developments taking place in software quality measurement research.

4.1.5.3 Testing For Security Evaluation - As a security evaluation technique, testing can stand alone or supplement other methods. Testing differs from other evaluation approaches primarily in that it searches for and evaluates flaws in the actual system (or application) as implemented, not as theorized or reflected in design and procedure documentation. As a result, it has the potential to detect implementation errors or even malicious software (which is not detectable by other means). It is also used to confirm or further expand upon findings from other forms of evaluation analysis. Finally, it can evaluate complex security factors not readily dealt with by other approaches, such as:

- Vulnerabilities which arise or are aggravated during stress loading.
- Actual difficulties in exploiting flaws.

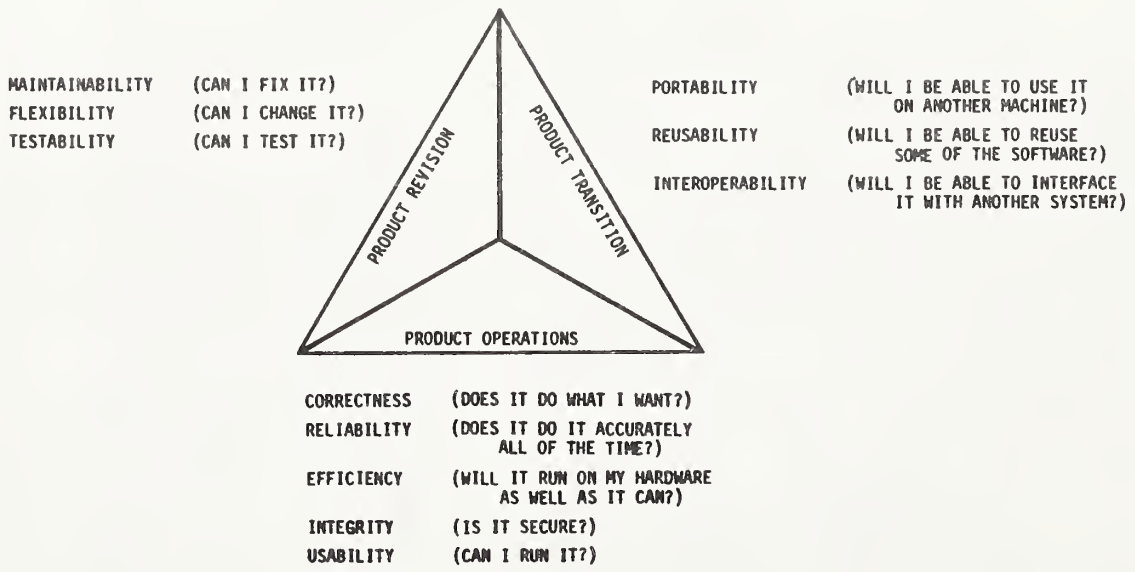


Figure 4-6.
Software Quality Factors [CAV78, p. 136]

Despite the continued development and improvement of evaluation methodologies, testing will remain crucial in confirming or expanding on their findings. Despite the great (though still basically unrealized) potential of such advanced techniques as security-kernel-based design and program verification, security testing will remain necessary for confirmation and added assurance. These advanced techniques may even have a synergistic effect in improving the quality of testing. For example, centralization of security software (e.g. in a kernel) would allow intensified use of program testing techniques; mathematically formal security requirements might be used to facilitate the derivation of test data. Security evaluation research cannot afford to ignore the testing area.

4.1.6 Other Approaches

The format for discussion of these approaches differs from the previous formats. The reason is that these approaches were not initially planned to be discussed, but were later determined to be of substantial value. Document preparation time limitations prevented more detailed coverage.

4.1.6.1 Canadian Institute Of Chartered Accountants [CIC75] -

Computer Audit Guidelines [CIC75] complements a document of Computer Control Guidelines [CIC70], both prepared by the Canadian Institute of Chartered Accountants. Members of the study group which prepared the report include R. Rosen, R. Anderson, L. Chant, J. Dunlop, J. Gambles, D. Rogers and J. Yates. Both books have been translated into French, German, Spanish, and Japanese. In addition, CICA offers a five-day seminar entitled "Computer Auditing" which is based on the audit text.

The document is included in this assessment because it contains a detailed, qualitative, structured control evaluation methodology. The evaluation process systematically assesses whether a set of generic control techniques is of sufficient quality to meet a number of minimum control standards. This document also incorporates checklists (as will be shown below). In terms of checklists, a session on administrative and physical controls at the 1977 NBS/GAO Invitational Workshop on Audit and Evaluation Computer Security [RUT77, p. 7-11] came to a consensus that this set of Audit and Control Guidelines was the "best single reference."

The overall objective of the document is to present "guidelines on the minimum standards and accepted techniques which should be observed in the audit of organizations using a computer". The preface notes that these guidelines are not the final word by stating "These studies are intended to stimulate thought, discussion and debate on matters of auditing theory and practice".

The evaluation process is structured around the control objectives shown in Figure 4-7. This figure is actually the summary page of the Control Evaluation Guide which is used to record the detailed control evaluation analysis.

Each control objective category (e.g. Processing) includes a number of high-level control objectives (e.g. L - "To ensure the completeness of data processed by the computer"). Each control objective, in turn, has been divided into from one to ten "minimum control standards" (e.g. L1 - "There should be some method of ensuring that all data is initially recorded and identified".)

These are further broken down into anywhere from one to ten related control techniques. Under the control objective of ensuring processing accuracy, for example, there are ten minimum control standards (e.g. M6 - "Controls must ensure that the accuracy of data is maintained during processing") and 28 generic control techniques (e.g. division of duties, control totals). Figure 4-8 is a portion of the Control Evaluation Guide which is used to assess how well the system meets its control objectives via its control techniques. It illustrates this control structure. Note that the sentence type changes from declarative (i.e. "there should be") to imperative ("assess") to reflect the active role of the auditor.

In performing the evaluation, the auditor would first proceed through the Control Evaluation Guide completing the Explanation column. Questions are answered in detail with yes/no answers typically being inadequate. For processing controls, relevant sections are completed separately for each major subsystem.

Step two of the process is to verify the techniques identified, summarizing the audit techniques used for this verification in the "Verification Techniques" boxes. The document provides guidance in selecting verification approaches for each specific control technique. Figure 4-9 illustrates. Down the left column are listed the generic control techniques. Across the top are listed three of the control objectives (L, M, and N) within the Processing category of controls. The numbered circles above each control technique indicate which minimum control standards are effected by the control technique. The figure also includes sample verification techniques. This records the steps performed during data collection and control evaluation. It constitutes documentation of a portion of the evaluation analysis.

Based on this verification, each control technique is then evaluated on the Control Evaluation Guide as being good, adequate, poor, or absent. From this evaluation of the individual techniques, the minimum control standard is then evaluated in the same way. Some small guidance is provided to assist in this evaluation. For example, an issue mentioned is "the extent to which the individual techniques are alternatives, overlapping, or complementary" [p. 176]. Primarily, though, the evaluation is left to the judgment of the auditor.

Control objectives		Summary of evaluation and major recommendations
I PRE-INSTALLATION	A Benefits of processing alternatives B Selection of facilities C Pre-installation plan	
II ORGANIZATIONAL	D Segregation of functions E Deployment of resources	
III DEVELOPMENT	F Benefits of processing alternatives G Development of effective systems and programs H Maintenance of systems and programs	
IV OPERATIONS	I Prevention or detection of accidental errors J Prevention or detection of fraudulent manipulation K Security against accidental destruction	
V PROCESSING	L Completeness of data M Accuracy of data N Authorization of data O Adequacy of management trails	
VI DOCUMENTATION	P Existence of adequate documentation Q Systems documentation R Program documentation S Operating and user instructions	

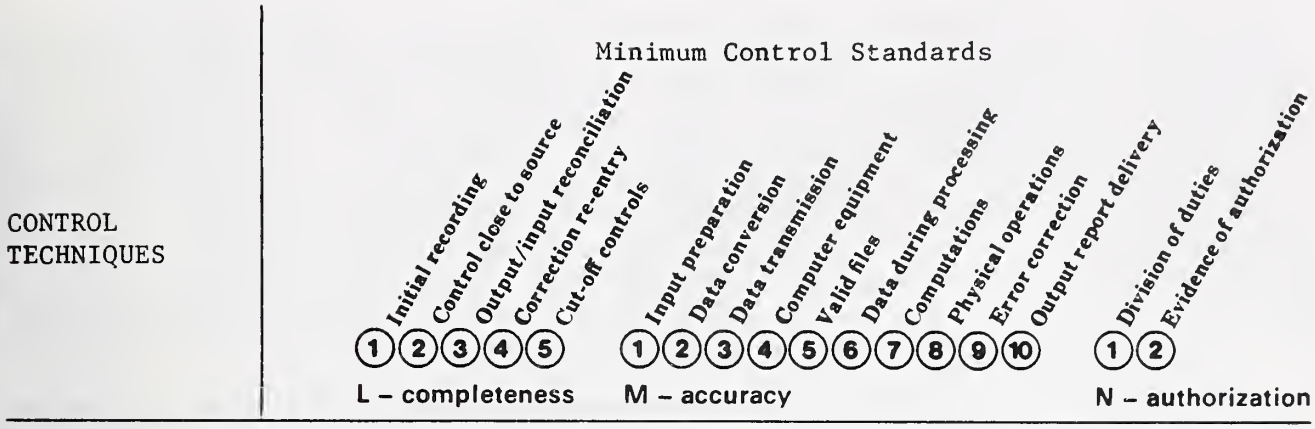
Figure 4-7
Control Evaluation Guide Summary [CIC75, p. 262]

N To ensure that all data processed by the computer is authorized.

		CONCLUSION				EXPLANATION	AUDIT	RECOMMENDATIONS
		good	adequate	poor	absent			
N1	**Assess the effectiveness of the separation of the EDP department from non-compatible functions within the organization, and of the segregation of duties within the EDP department.**						Compensating audit procedures	
N1-1	*There is a separation of the functions of (i) initiation and authorization of transactions; (ii) recording of transactions; and (iii) custody of assets.*						Verification techniques	
N1-2	*Access to critical forms is restricted to individuals responsible for the initiation function.*							
N1-3	*Access to the computer and computer files and programs is restricted to designated employees.*							
	Other:							
N2	**Assess the methods used to ensure that only authorized data is processed and that input documents bear evidence of authorization and are reviewed by the control group for such evidence.**						Compensating audit procedures	
N2-1	*In a batch processing system, clerical procedures are used to authorize input and to subsequently scrutinize it for proper authorization.*						Verification techniques	
N2-2	*To the extent practical, computer routines are utilized to authorize input and subsequently scrutinize it for proper authorization.*							
	Other:							

Figure 4-8
Control Evaluation Guide Excerpt [CIC75, p. 286]

Reprinted with permission from Computer Audit Guidelines, 1975, published by the Canadian Institute of Chartered Accountants, Toronto, Canada.



① ② ③ ④ ⑤ ① ② ③ ○ ○ ○ ○ ⑧ ⑨ ⑩ ○ ②

6. Control group

Control group receives input, verifies authorization, reconciles processing, distributes output, and ensures errors corrected. (See Control Technique *I1-1*)

BASIC VERIFICATION TECHNIQUES

Determine through discussion and observation, the existence, terms of reference and independence of the control group.

Ascertain, through examination of control logs and other documentation, that the control group scrutinizes all input and re-entry documents for completeness.

Examine procedures relating to rejected items (errors) to determine that they are adequately logged and that provision is made to ensure re-entry after necessary corrections have been made. (See Technique No. 27 — Error log)

Determine, through discussion and observation, that the control group ensures that all output is properly distributed.

ADDITIONAL MANAGEMENT VERIFICATION TECHNIQUES

Determine that the terms of reference and established practices of the control group are adequate and relate to the size of the installation.

Particular attention should be directed to ascertaining that the control group cannot be bypassed, except in circumstances subject to prior approval and, that in such cases, adequate alternative arrangements have been made for the performance of the necessary control activities.

○ ○ ○ ○ ○ ① ② ③ ○ ○ ○ ○ ○ ○ ○ ○ ○

7. Self-checking digit

Self-checking digits used on key codes not otherwise controlled.

BASIC VERIFICATION TECHNIQUES

Determine, through observation, that self-checking digits are being used on key codes not otherwise controlled.

Ascertain, through the use of test data, that they are being effectively employed, either at the data conversion stage or as a computer edit routine.

Figure 4-9
Verification Guide Excerpt [CIC75, p. 211]

Reprinted with permission from Computer Audit Guidelines, 1975, published by the Canadian Institute of Chartered Accountants, Toronto, Canada.

Next, non-processing controls such as manual controls or controls from other checklists are evaluated. The reason given for this step is that the control objectives and techniques listed are specific to computer-based systems. It is not clear, however, that "traditional manual system controls" warrant separate handling. For example, an example given of a manual system control is: "Credit notes for returns and allowances should be approved by an employee who has no access to cash receipts or other company funds" [p. 176]. This would seem readily encompassed by the general need for segregation of duties. It would seem both desirable and feasible to eliminate this step from an implementation of this methodology.

The final step is to plan compensating audit procedures to offset areas of control weakness. In doing this, an analysis of control deficiencies is made to determine specific needs for compensating controls. Figure 4-10 illustrates the nature of the guidance provided in doing this. Notice that the "possible deficiencies" and "consequences" are very similar to vulnerabilities (as in questionnaires such as the AFIPS checklist discussed below) and exposures (as in the Touche Ross methodology). It is unusual that such detailed consideration of vulnerabilities and exposures is performed only after the control evaluation. It would be a simple and perhaps desirable revision to include the analysis of control deficiencies as a part of the verification step (preferably after any document reviews but before any testing).

4.1.6.2 Arthur Andersen & Co. [AAC78] - Arthur Andersen has developed a "Guide for Studying and Evaluating Internal Accounting Controls" which has gained prominence both for its discussion of control objectives and its presentation of a transaction flow review approach to evaluation. The document deals with financial controls and is oriented around the "cycles" of a business activity. The concept of cycles will be described here to provide a background for discussion of the evaluation approach.

A cycle is a group of similar economic events (e.g. treasury,[19] expenditure, conversion, revenue, financial reporting). Cycles are defined "to categorize the flow of economic events, since this is consistent among entities, rather than the flow of accounting information, which is not consistent" [p. 33]. Cycles "provide a meaningful framework for viewing and studying a business and its accounting processes without being overwhelmed by the details of systems, procedures, techniques, and processing methods" [p. 33]. The economic events comprising a cycle are converted into transactions for processing through an entity's accounting systems. This then, represents the organizational structure upon which the evaluation approach is applied.

SYSTEM EVALUATION METHODOLOGIES
SECURITY EVALUATION METHODOLOGIES

TECHNIQUE	L – completeness: Initial recording (1)	Control close to source (2)	Output/input reconciliation (3)	Correction re-entry (4)	Cut-off controls (5)	M – accuracy: Input preparation (1)	Data conversion (2)	Data transmission (3)	Computer equipment (4)	Valid files (5)	Data during processing (6)	Computations (7)	Physical operations (8)	Error correction (9)	Output report delivery (10)	N – authorization: Division of duties (1)	Evidence of authorization (2)
			○							○	○		○				
20. Software controls (OS; IOCS) Operating systems; input/output control systems, etc.			○							○	○		○				
21. Library Control over files.			○							○							
22. Control over programming											○	○	○				

20. Software controls (OS; IOCS)
Operating systems; input/output control systems, etc.

POSSIBLE DEFICIENCIES

Read-after-write checks are not employed in all appropriate places.

Address-compare checks are not utilized on all storage transfers.

Software system does not always include checking of internal labels or this check is sometimes bypassed.

Full utilization is not being made of operating system features.

CONSEQUENCES

Errors may be introduced during processing or during transfers of data to or from machine-sensible files.

Mispostings or up-dating of wrong master file records may occur (such errors are not generally detectable by control totals).

Incorrect files may be processed inadvertently resulting in certain transactions effectively remaining unrecorded (e.g. when an outdated file is mistaken for a current one), or certain transactions may be recorded twice, or recorded incorrectly, or misclassified.

Operator errors may be more frequent (because of inadequate monitoring by operating system) leading to undetected inaccuracies in processed data.

21. Library
Control over files.

POSSIBLE DEFICIENCIES

Library is not maintained to control files.

Library exists but its operation is inadequate (which may result in authorized requests not being required for file issue, or not being based on the production schedule, or no log of file usage being maintained).

CONSEQUENCES

Incorrect files may be processed inadvertently resulting in certain transactions effectively remaining unrecorded (e.g. when an outdated file is mistaken for a current one).

Certain transactions may be recorded twice or recorded incorrectly or misclassified.

Unauthorized access to files may lead to deliberate inaccuracies or fictitious transactions being inserted in processed data.

22. Control over programming

POSSIBLE DEFICIENCIES

Poor programming controls permit logic errors to occur in programming.

CONSEQUENCES

Undetected logic programming errors may cause inaccuracies in processed data.

Figure 4-10
Control Deficiency Guide Excerpt [CIC75, p. 233]

The Arthur Andersen & Co. approach to evaluation of internal accounting controls consists of four tasks:

- o General risk analysis
- o Transaction flow reviews
- o Evaluation of internal control techniques
- o Compliance testing

In the risk analysis, financial planning and control are reviewed to document the processes and obtain an overview of the internal control environment. Most of the document consists of illustrative control objectives and "risk identifiers" which presumably assist in the risk analysis process. These will be briefly discussed. The document defines 117 illustrative control objectives for the business cycles. These cycle-oriented objectives are derived from several high-level systems control objectives. They are grouped as follows:

- o Authorization objectives
- o Transaction processing objectives
- o Classification objectives
- o Substantiation and evaluation objectives
- o Physical safeguard objectives

Within this grouping, they are further subdivided into subgroups applicable to each business cycle.

From this structure, a group of "risk identifiers" is formulated. There are 35 risk identifiers analyzed for over 100 pages. Basically a risk identifier is defined by a logically related group of objectives. Each one contains a discussion of the objectives, examples of techniques (i.e. controls) that might be used to achieve the objectives, and risks (i.e. exposures/impacts) if the objectives are not achieved. (For those who prefer the analysis to be more rigidly structured around the business cycles, an appendix is provided which organizes the objective/technique/risk data around the cycles rather than around the grouping described above.)

To the extent the Arthur Andersen "general risk analysis" task involves review or formulation of risk identifiers, it is indeed a risk analysis in the traditional sense. While there is no explicit treatment of threats or assets, there is explicit, though not quantitative, treatment of control objectives, controls, and impacts. The inclusion of control objectives in such a risk analysis is discussed in section 4.3.3.2. In summary, the general risk analysis will provide a foundation for the more detailed tasks which follow.

After the general risk analysis is completed, the second task is

---

[19] The treasury cycle is somewhat distinct in that it provides the capital for the other cycles. It also handles cash management, including outside investments.

to perform transaction flow reviews. The basic idea is to follow a transaction through its processing and identify (not evaluate) the controls applicable at any point. As described in the document:

> "A transaction flow review is a detailed study of the entity's internal accounting controls over a particular category of transactions.... The purpose of such reviews is to obtain information concerning the entity's cycles, transaction flows, functions performed within each transaction flow, and the control techniques employed to prevent, detect, and correct errors and irregularities. In addition, as part of a transaction flow review, cycle control objectives applicable to the functions should be identified" [p. 187].

The typical mechanism for performing the review is a flow chart.

> "After determining the flow of transactions in a function and documenting such flow in a flow chart, the reviewer identifies the cycle control objectives that are applicable to the function and notes the specific internal control techniques used by the entity to provide reasonable or partial assurance that such objectives are achieved. Each control technique noted should be identified with a specific objective.... Internal control techniques should be distinguished from processing steps" [p. 190].

This type of analytical approach is currently very popular with the "Big Eight" accounting firms. The American Institute of Certified Public Accountants (AICPA) published a report similar to (but smaller than) Arthur Andersen's which both recommends internal accounting controls and describes a transaction flow technique [AIC79, pp. 23-24]. Interestingly, a transaction flow approach was also recommended at an NBS/GAO invitational workshop as the desired way to perform a data communication audit [RUT77, p. 10-4, in Section 10, "Audit and Control of Data Communications Networks - A Consensus Report"]. At a subsequent NBS/GAO workshop [RUT80], transaction flow analysis was noted as a reasonable way to evaluate application systems. (See Session 7: "An Approach to Identification and Audit of Vulnerabilities and Controls in Application Systems".) The key to Transaction Flow Auditing (TFA), according to Mednick, is that it "concentrates on the 'why's' of internal control and not just the 'whats' [MED79, p. 61].[20]

> "TFA is a new and different way to look at internal accounting controls. The traditional way has been to compare them to long lists of control techniques and rely on intuitive judgment to identify the important ones. This method has become inadequate, however, as systems have grown in complexity. It has led to excessive controls in some areas and insufficient controls in others. The focus has been placed, mistakenly, on the techniques actually in use rather than the reasons they are necessary. The TFA

> approach is different in that it concentrates, instead, on the need to meet specific control objectives . . . . This is probably the most significant aspect of the TFA approach" [MED79, pp. 58-61]."

Another advantage of TFA is that it allows the reviewer to "localize" and focus attention. The resultant analysis therefore becomes more manageable and meaningful.

Similar "focused" analyses would seem useful in the security community where the objective is more on other aspects of security than data integrity and where emphasis is often placed on exception conditions and anomalous cases. Indeed Arthur Andersen claims to have modified the transaction flow approach in order to tailor it for reviews of other than integrity and used it with substantial success. The modified methodology is proprietary, however.

The remaining two tasks in Arthur Andersen's approach to evaluating internal accounting controls are evaluation of internal control techniques and compliance testing. Little guidance is given in either area (i.e. a total of two pages for both).

4.1.6.3 AFIPS Security Checklist For Computer Center Self-Audits [AFI79] - This "checklist" presents an excellent basis for a detailed security evaluation. The structure, comprehensiveness, and overall quality of the document are outstanding. With supplemental methodological guidance, this document would constitute an excellent detailed guideline on security evaluation. It is important to note that this is a security checklist and is oriented around security concerns, not audit or risk concerns (as discussed further in Section 5.3).

The idea of the need for a System Review Manual originated in a special AFIPS Systems Certification Workshop held in 1971. The conclusion was that if "system certification" could be addressed, it would be best to publish a series of manuals on "preferred practices" [LOB80, p. 11]. However, later on in that same year, a second, AFIPS-sponsored workshop concluded that "We weren't yet ready to 'certify systems' and that preferred practice manuals were fraught with problems.... [but] it was possible to develop checklists to probe and reveal the important qualities and characteristics of a system [AFI74, back cover]". The workshop recommended a series of System Review Manuals as the logical first step with the subject of the first manual being privacy and security.

Work on the manual was begun by Mary Elizabeth Stevens in mid

---

[20] Note the similarity to the CICA emphasis on the "point of incidence of the error to be prevented" rather than the "point of incidence of the control" (see Section 3.1.4).

1972. The draft was reviewed by an AFIPS committee and under the authorship of Robert Patrick underwent extensive revision before initial publication in late 1974. The current version, published in 1979, represents the product of still another major revision authored by Peter S. Browne.

The document consists of over 1,000 "embarrassing questions" relating to all facets of ADP operations and is equally applicable for systems, applications, and installations. The questions are divided into nine major sections: Planning and Risk Analysis; Physical Security; Backup and Recovery; Administrative Controls; Systems Hardware and Software; Communications; Distributed Risk; Applications (including integrity and development); and Security Audit. These are expanded upon in Table 3-4. Each section is introduced by several pages of discussion of terminology, concepts, available technology, and recommended practices. Then follows a detailed checklist (set of questions) forcing the reader to consider a multitude of disturbing and unlooked-for possibilities. [The following quote is on page 13.]

> "Each set [of questions], and sometimes each series within a set, is introduced by an 'imperative'. The imperative is a working statement designed to tell the author or evaluator the tasks to be accomplished based on the associated material. Each imperative is accompanied by a 'primary issue' which states the objective of the set of questions taken as a whole.
>
> The questions are organized according to functional areas, with topical subdivisions if necessary. Blocks of questions can be conveniently parcelled out to members of the evaluation team. In addition, the questions are categorized into low, medium, and high risk environments. These three levels are determined by evaluating the potential dollar loss or recovery time. One successful approach has used the following guidelines:
>
> - Low risk - potential losses amount to $10,000 a year or less; recovery time, 30 work days.
> - Medium risk - potential losses are between $10,000 and $100,000; recovery time, up to 100 work days.
> - High risk - potential losses are greater than $100,000; recovery time, in excess of 100 work days.
>
> Of course potential losses should also be related to the size and value of assets, perhaps expressed as a percentage. A loss greater than 10 percent of an organization's assets, or one year's profit, might be classified as a medium risk situation.

> The low risk category is considered to be the minimum level of control necessary for most organizations, large or small. However, even here, the question of judgment is necessary. High risk in one installation may be a basic environmental hazard in another; therefore all questions should be reviewed once the basic threats have been enumerated. Only an analysis of data sensitivity . . . will determine the real need for controls."

The document does not intend to be a "security cookbook". It "is intended to provoke thought.... it is a structured aid rather than an all-inclusive plan" [p. 11]. Brief guidance is given on how to arrive at a composite security evaluation based on responses to the checklist. Three types of evaluations are defined (i.e. internal evaluation by internal personnel; internal evaluation by external personnel; and external evaluation by external personnel) with advantages and disadvantages listed for each. A team (rather than an individual) approach to evaluation is recommended. Guidance is also provided on the conduct of the evaluation review.

Despite this guidance, the document is primarily intended to be a checklist, not a methodology. Its main values are in heightening security awareness and ensuring a complete review of potential vulnerabilities and controls. Other security checklists may approach this one in completeness, but they typically cost $100 - $500 (versus $25 - $35) and are often poorly structured. (One exception is the checklist summarized in the next section.)

In SDC's experience, one military organization had spent a substantial amount of time preparing a security checklist as thorough as possible for use in their internal evaluations. On being introduced to the AFIPS checklist, they were very impressed by its completeness and incorporated it into their security program. It includes a number of issues which the organization's checklist had overlooked.

One of the generic steps in the security evaluation process is the listing of security controls. A key objective of this step is to be as complete as possible. This checklist would be a valuable addition to any of the methodologies listed above or below in achieving this objective. Most methodologies, of themselves, provide little assistance in this area.

Many of the controls discussed in the document are not available on existing systems. This is more a failing of existing systems than the document. Naturally, it is possible to envision subtle controls which had not been included such as photographing visitors (just as stores occasionally photograph customers who pay by personal checks) or accommodating confinement channels[21] [LAM73]. The absence of such controls does not significantly detract from the value of the checklist. The true issue in ensuring an acceptable level of completeness is the need to periodically update a checklist to accommodate new technology and ideas from other checklists. For

example, one area of new technology which has not been given thorough treatment in the AFIPS checklist (in light of recent technology advances) is the area of encryption which includes such issues as key management.

In summary, the "overriding objective" of the document is to permit readers to "evaluate controls", and thereby inexpensively discover potential vulnerabilities in their systems. It serves this objective well.

4.1.6.4 Internal Controls For Computerized System [FIT78] - Written by Dr. Jerry FitzGerald, this document is a checklist of over 650 controls. They are organized into nine control groups:

- o General organizational controls
- o Input controls
- o Data communication controls
- o Program/computer processing controls
- o Output controls
- o On-line terminal/distributed systems controls
- o Physical security controls
- o Data base controls
- o System software controls

Before listing the specific controls in each general group, the document lists concerns/exposures (such as program errors, unauthorized program changes, security/theft, and error handling in the program/computer processing controls matrix, p. 37) and resources/assets (such as application programs, data record integrity, and central system in the same matrix). For each control group, the relationship between concerns/exposures, resources/assets, and controls/ safeguards are summarized in a matrix. This approach allows readers the advantage of immediately focusing their attention on only those areas they feel to be of primary concern to their situation.

The document explicitly states that the matrices "do not comprise a methodology on the conduct of an internal control review. Instead, the overall methodology on how to conduct an internal control review is assumed to be already established within the organization" [p. 4]. NBS SP 500-19 includes some insights in the use of the data communication control matrix in security reviews. For example, it notes that the matrix should be used to review security in light of each of the specific applications using the network [RUT77, p. 10-6]. This matrix approach has been integrated into a methodology for

---

[21] This has become a subject of significant concern in the DoD security community. Neugent [NEU80,pp. G-1 through G-17] presents probably the most complete summary of confinement issues.

evaluating the effectiveness of proposed controls in systems under development in a newer work by FitzGerald[FIT81].

While not quite as extensive as the AFIPS checklist, this document would also be of value as a supplement in ensuring thorough analysis of controls in a security evaluation.

4.1.6.5 Coopers & Lybrand [C&L82] [HAL85] - Coopers & Lybrand (C&L) has an integrated audit approach and methodology of long standing for ensuring their clients' systems are secure and well-controlled. They have documentation of this approach that takes the auditor through a step-by-step review of the security and controls on both an application level as well as the general (integrity) controls level. Their integrated audit approach was developed to address complex systems that need control reviews and to provide effective allocation of audit resources. The characteristics of this integrated audit approach are that it is comprehensive, systematic, standardized, and provides a systems-based approach to control evaluation. It is a proprietary methodology.

The components of this audit approach are:

1. Planning
2. Gaining an Understanding of the System
3. Recording the Understanding
4. Confirming the Understanding
5. Evaluation of Internal Control
6. Compliance Testing of Controls
7. Audit Response to Control Weaknesses
8. Substantive Procedures
9. Findings and Recommendations (the Audit Report)

Each component is expanded upon in the documentation and requires the use of several exhibit forms by the auditor. For example, in doing an Evaluation of Internal Control, emphasis is placed on the following control objectives:

- data is completely input, processed, and updated to appropriate files

- o data is accurately input, processed, and updated to the appropriate files
- o data input is authorized in accordance with management's intentions
- o data is correctly and currently maintained on master files.

The application controls and programmed procedures necessary to achieve these objectives are then described. Finally, the use of their Computer Internal Control Questionnaire (CICQ) (which addresses evaluation of controls) and Control Matrix (which summarizes CICQ responses) leads to their Record of Control Weaknesses document. The C&L Internal Control Reference Manual assists in the evaluation of controls by defining what constitutes adequate control procedures and describing the impact of a control deficiency.

C&L uses this integrated audit approach for application or integrity (general) control reviews, system development life cycle, review of standards, security reviews, and operational audits. This methodology is under the jurisdiction of the C & L Computer Audit Assistance Group (called CAAG). The Handbook of EDP Auditing [HAL85] outlines this approach in detail giving practical examples of use as well as audit guidelines for different types of systems. It also includes sample documentation which supports this approach. This 1100 page plus book with 32 chapters includes chapters on Security, Microcomputers, Data Bases, Computer Abuse, Systems Development and On-Line Systems, all of which cover aspects of security and control over access. CAAG also does extensive training in the EDP audit field. For both clients and non-audit clients, this training includes comprehensive case studies utilizing the methodology in all related forms and documentation aids on a sample organization.

C&L's library of audit software, available on mainframes, minicomputers, microcomputers, and through timesharing includes software to address the integrity controls over systems. These integrity or general controls are:

- o Implementation Controls
- o Program Security Controls
- o Computer Operations Controls
- o Data File Security Controls
- o System Software Controls

Where software packages are available from vendors that specifically address security (RACF, ACF2, and TOP SECRET), C&L has developed audit guides to assist the auditor in reviewing the implementation of these packages. With other vendor software such as IMS DC and CICS, C&L has developed software to assist in reviewing the

security implementation capabilities imbedded in these systems. The audit software is available by special arrangement from C&L for use by others.

C&L develops guidance materials as part of its National Office R&D Function on Technology. Topics addressed in 1985 that are relevant here include End User Computing (Controls), Micro to Mainframe Links (Controls over Uploading and Downloading of Data), and an access model that addresses the risk for data security, confidentiality, and privacy.

The security review performed by C&L is a four phase approach:

1. Review and Evaluation - determines the strengths and weaknesses of the existing security and control structure within the organization. Security procedures, policies, and practices are evaluated to determine their completeness, intent, and effectiveness.
2. Exposure Assessment - identifies the risk to which the organization is exposed as a result of specific weaknesses identified in the first phase. Risks are typically classified as: destruction of physical assets or information; disclosure of confidential information; removal of physical assets or information; corruption of data or programs; and interruption of service. The impact of events are determined using two factors, the frequency of occurrence and the amount of the potential loss.
3. Preparation of Recommendations - recommendations include procedures that should be implemented to mitigate the exposure and its potential impact on the organization.
4. Implementation Plan - a strategy for implementing the recommendations developed in phase three. The implementation plan consists of identifying the sequence and timing of activities, the nature and extent of the resources required, and the estimated cost of implementation.

4.1.6.6 Auditing Computer Systems [PER81] - This is an excellent reference work for anyone entrusted with establishing an auditing program. Published by the Faim Technical Library, it is a three volume set of looseleaf binders. William Perry is listed as principal contributor. Since this technology assessment is not concerned with the general practice of auditing (only the security evaluation component), the overall document will not be discussed here. Suffice it to quote from a review by Donald Adams in EDPACS [ADA78, p. 16]: "This comprehensive work deserves a prominent position in any serious library of EDP audit material. Many auditors may find this to be the most useful volume they have ever acquired."

Worthy of mention here is a section on the review of internal controls in Chapter 5 (Auditing an Application). In the space of a few pages is presented a methodology for evaluating individual controls and deriving a quantitative overall application security evaluation. While the methodology is admittedly simplistic, ignoring such basic factors as weightings of the controls, interrelationships between them, and so forth, it addresses most of the evaluation steps discussed in this technology assessment. The approach uses two tools, an internal controls checklist and a control evaluation work sheet. The evaluator proceeds through the checklist, rating each control as strong/good (5), adequate (3), or weak/poor (1). Then, based on the arithmetic mean, the application is rated. Scores are as follows:

1.0	-	2.75	Internal control is generally weak.
2.76	-	3.25	Internal control is adequate to the job.
3.26	-	4.0	Ideal level of internal control.
4.01	-	5.0	Superior level of internal control, but may not be cost effective.

A two volume work by William Perry, entitled "Internal Controls" [PER80] comprehensively discusses internal controls from the point of view of design, maintenance, and assessment in the first volume and then systematically describes hundreds of controls in terms of a standardized set of characteristics in the second volume.

To the extent simplicity is desirable, an approach such as this might be of use. Intuitively, however, the dangers incumbent in such an approach could be significant. Certainly all controls are not equally important. Just as certainly, the absence of a few key controls could offset the presence of many others. While the accuracy and use of this methodology may be debatable, its simplicity is not. The fact that such a simple technique is suggested at all is worthy of note.

4.1.6.7 Information Security Handbook [WIL80] - This is an information security handbook for internal auditors. The handbook "provides a comprehensive information-security program, checklists, and an audit approach to assessing the effectiveness of the information security program within an organization. Its scope is limited to information security, the prevention of information from being disclosed to an unauthorized recipient" [WIL80, p. viii]. In the author's view, this handbook provides a complete approach to auditing the security of all proprietary information wherever it is resident in an organization. It is not limited to security of the data center.

The handbook provides guidance in establishing and auditing an information security program. In the audit area, it does not present a detailed approach but rather addresses overall structural issues. It also provides hints on the application and use of detailed audit approaches which might be formulated from its guidance. It contains twenty checklists, some of them lengthy and some very short, which are strictly oriented towards information security. Examples of the checklist topics are: Classification of Information; Islands of Security; Top-Priority Document Control; New Product Security; Trade Secrets; Intellectual Security; Remote Computing; New System Design, Development, Test, and Implementation; Auditability; and Encryption.

4.1.6.8 Department Of Health And Human Services [HHS78] [HHS82] - In 1978 HHS developed an extensive ADP Systems Security Manual oriented around checklists which was used for security evaluation. The document defined the overall HHS security program and included ADP security principles, responsibilities, and authorities. It also included Risk Management guidelines. Policy, mandatory administrative and physical safeguards, and a security requirements checklist were provided in each of the following areas:

- o ADP Application System Design and Development
- o ADP Application System Users
- o Computer Facilities
- o Mini-/Micro-Computer Facilities
- o Remote ADP Work Stations
- o Ancillary Facilities
- o Telecommunications
- o Grants, Contracts, and other agreements

The checklists were supplemented for internal audit purposes with expanded checklists which are not externally releasable. Evaluations were usually performed by the facilities and systems managers themselves although external evaluation teams were sometimes used (comprised, for example, of Inspector General, physical security, or personnel security representatives).

In 1982, using their 1978 document as a foundation, HHS revised their manual extensively [HHS82], orienting the new document around a matrix of minimun security requirements and safeguards. Facility type (e.g., large, small, remote, ancillary, office automation) and application system criticality/sensitivity level are taken into consideration in drawing up this matrix. Managers of computer systems are expected to incorporate the minimum security safeguards in their facilities and applications until a security review indicates the need for more specific controls.

In the document's discussion of an automated information systems security management program, the security policy, authority sources, personnel responsibilities, facility and application categorization, and key definitions are explicitly covered. The specific areas treated by the security program for which detailed guidance is given are:

1. Risk Management
2. Personnel Security
3. Information Processing Applications
4. Data Communications
5. Operating Systems
6. Physical Security
7. Acquisitions, Grants, and Cooperative Agreements
8. ADP Security Program Review and Evaluation

The document also contains forms for (1)ADP Security Resource Inventory, (2)ADP Security Financial Plan, and (3)ADP Security Annual Plan and Quarterly Progress Report. HHS has also drawn up a draft document, "Audit Guide for Review of Security Over ADP Facilities," for use by their security audit personnel.

4.1.6.9 Department Of Agriculture [DOA80] [DOA84] - The Department of Agriculture (USDA), in 1980, tried formulating a security evaluation methodology for use in complying with the certification requirements of OMB A-71, TM-1 [OMB78]. The methodology was to have two parts, a series of questionnaires developed internally by USDA and methodological guidance, developed by a contractor.

There were three distinct questionnaires, structured as follows:

- o Questionnaire on Management Policies and Procedures
  - Personnel
  - Security
  - Contingency Plan
  - Regulations and Standards
  - Application Requirements and Approvals

- Financial

o Questionnaire on Application Controls and Security Provisions

- Processes
  . input
  . processing
  . storage and transmission
  . output
  . error correction

- Practices
  . separation of duties
  . security
  . contingency situations
  . working procedures
  . dissemination of information
  . authorization/approval
  . testing/maintenance

o Questionnaire on Computer Operations

- Application
- Facility and Operations
  . security
  . electrical devices and power supply
  . fire prevention and detection
  . operating procedures
  . systems software
  . systems access control
  . media storage
  . backup

The questionnaires were based on an analysis of numerous checklists. (All were examined in this technology assessment except one which is now out of print.) A rule of thumb underlying construction of the questions was that they should be phrased such that a yes or no answer would not suffice. The intent here was to help ensure that the appropriate level of complexity was included in the evaluation. One of the preliminary findings of USDA in formulating these questionnaires was that evaluations will take more time than was originally anticipated.

The methodological guidance being developed would have placed particular emphasis on interview techniques, on the presumption that the manner in which the questions are asked can be as important as the questions themselves. When the security evaluation methodology had been developed, it was to be tested with the USDA agencies.

Several potential implementation problems were recognized by USDA. First, although it is crucial to key an evaluation around specific management objectives, this can be difficult because "management" may represent several individuals with differing needs. Second, obtaining funding for implementation of the evaluations might be difficult. Lastly, given the cost and difficulty of retrofitting existing applications, it was seen as likely that the methodology would be of more importance to new applications.

In order to overcome the problems of pesonnel and cost foreseen in pursuing the above centralized approach to evaluation, the USDA decided to place their emphasis on security requirements for their agencies and allow each to do evaluations on their own. They have, therefore, rewritten their security standards and placed them in a new document "ADP Security Manual" [DOA84]. Brief guidance on risk analysis (which conforms to [FIP65]) and certification/recertification (which conforms to [FIP102]) were placed in this new document. The USDA agencies are now being referred to the GAO "Black Book",[GAO81-1] for detailed evaluation questionnaires. USDA has also written a security policy document to complement this new security standards document. The hope is that this approach will be a more cost-effective one for the Department.

4.1.6.10 GAO Audit Guides [GAO81-1] [GAO81-2] - In June, 1981 GAO published two audit guides: (1)"Assessing Reliability of Computer Output" [GAO81-1] and (2)"Evaluating Internal Controls in Computer-Based Systems" [GAO81-2]. These guides are designed to address two different but related audit situations. [GAO81-1] provides guidance to generalist auditors on the appropriate steps to be taken when using computer produced data in their evaluations. Depending on how the data is being used and the deficiencies encountered, the auditors may recommend a complete review of the system that produced the data. In other instances, the audit itself may be to review the accuracy and reliability of a particular system or application. In either case, [GAO81-2] provides detailed guidance for information system evaluations that follow.

The reliability guide [GAO81-1] advises the auditor to perform the following four steps, stopping after each step to ask whether reliability of that data is still an issue needing further investigation. The four steps are:

1. Identify computer data that will be used in the audit.

2. Determine the importance of the data to the audit.

3. Determine the source of the data and understand its flow through the system.

4. Conduct brief tests for data reliability.

Each step is discussed and then condensed into a set of audit procedures at the end of which the auditor's options for either accepting the risk or continuing with the reliability assessment are outlined. In Step 2, review of prior audits and evaluations is the central activity. In Step 3 the preparation of a document flow diagram, the verification of the document flow accuracy and completeness, and determnation of availability of corroborative data for testing are the main activities. In Step 4, suggestions for testing reliability are given (e.g., confirmation with individuals who are independent sources, review of data for reasonableness, comparison of data with independently arrived at values). A User Satisfaction Questionnaire is also provided to assist in reliability assessment. If this testing suggests critical data is indeed questionable, the auditor, based on certain criteria, may either just report this and end the review or decide to go to an internal controls review as outlined in [GAO81-2].

The second GAO audit guide [GAO81-2] provides a structured approach that "helps evaluate the total system from origination of source documents to final distribution of output products. Primary emphasis is placed on assessing the system's or application's reliability in processing data in a timely, accurate, and complete manner." This guide is divided into four phases as follows:

1. Background Data Collection

2. Internal Controls

3. Detailed Analysis and Testing

4. Reporting

Background data on the system in question is collected via three included checklists: one on the agency, one on the ADP department, and one on the computer application. Using the information obtained through the checklists, the auditor can then ascertain the status of internal controls by filling out questionnaires. There are three questionnaires that address aspects of top management controls, five questionnaires that address aspects of general controls over the data processing function, and four questionnaires that address controls over the computer application. After each group of questionnaires is

completed, a profile of the adequacy of each of these three groups of internal controls (i.e., top management controls, general controls, and application controls) is drawn up. The profile form also calls for a judgment on the part of the auditor, on the level of potential risk in each aspect of each of the three groups of internal controls. The guide then provides, for each of the three sets of internal controls, a matrix of additional audit steps needed for control areas that were judged to have a medium to high level of risk.

The Detailed Analysis and Testing phase then carries out the additional audit steps indicated in each of the three matrices. These additional steps may include preparation of a data flow diagram, observing manual and automated data processing, obtaining further information on user satisfaction with computer output, and/or verifying master records with source documents. For areas of disclosed weakness, standard computer audit techniques (e.g., test decks, specialized software packages) may be used to determine the amount and extent of errors that exist. The guide then briefly describes the Reporting phase which may document discrepancies or recommend that additional audit work be done. GAO is in the process of revising these guides at this time.

4.1.6.11 Department Of Energy [DOE83] [DOE84] - The Department of Energy (DOE) has very actively responded to the Office of Management and Budget (OMB) regulation on internal control (OMB Circular A-123 [OMB81,83]), to the regulation's legislative enactment (the Federal Manager's Financial Integrity Act of 1982 [FIA82]), and to the OMB regulation on computer security (OMB Circular A-71, Transmittal Memorandum #1 [OMB78]) [22] by issuing two directives on policies and standards (one on computer security in 1979 and one on internal controls in 1984) and two very pertinent documents. The two documents are:

1. ADP Internal Control Guideline, August 1984 [DOE84]

2. Draft Guide for Performing Internal Control Reviews, April 1983 [DOE83]

The more recent document [DOE84] is the main statement of the DOE approach to computer security and internal control. This document discusses in depth in Section VI the requirements of the two DOE directives and then compares these requirements in order to determine the relation between internal control and computer security activities. This comparison then enables DOE to draw up a consistent

---

[22] For a brief discussion of this law and these OMB regualtions see Section 1.1.1.

program for internal control and computer security. Some of the major points concerning this comparison that are relevant here are:

1. There are three levels of control and they are applicable to both ADP internal control and ADP security. These are management controls, operations controls and application controls.

2. Conceptually, ADP security is a subset of or component of ADP internal control [DOE84, p. 18] [23]

3. "...vulnerability assessments [required by OMB Circular A-123] rank programs and functions for conducting subsequent internal control reviews while risk analyses [required by OMB Circular A-71 TM1] rank the degree of risk so security resources can be properly apportioned." [DOE84, p. 23]

4. "...vulnerability assessments focus more heavily on management and the general control environment"..."while risk analyses evaluate threats and attempt to eliminate them." [DOE84, p. 23]

5. ADP internal control review (required by OMB Circular A-123) "encompasses a detailed examination of ADP internal controls to determine whether adequate controls exist and are functioning as intended." [DOE84, p. 23]

6. Security audits (required by OMB Circular A-71, TM1) include "the written recertification of the adequacy of protection of each operational computer application that contains sensitive data." [DOE84, p. 24]

7. Although internal control reviews and security audits both evaluate the existence of adequate protection, internal control reviews "analyze and document the overall process of ADP management, organization and operations in the context of the general control environment and of formulated control objectives" while "security audits primarily focus on the protection adequacy of each operational computer application that contains sensitive data." [DOE84, p. 24]

8. "Internal control reviews are direct extensions of the vulnerability assessments, while there is not necessarily a direct relationship between the risk analyses and security audits." [DOE84, p. 24]

It should be noted here that, in the terminology of FIPS PUB 102,

---

[23] Although the DOE definition of computer security, addressing mainly safeguarding ADP resources, is somewhat narrower than the NBS definition, the NBS definition still makes security controls a subset of internal controls.

vulnerability assessments are the equivalent of a high level basic evaluation procedure and internal control reviews are the equivalent of detailed evaluations.

Using this framework for the relation between internal control and computer security, DOE's ADP internal control guideline [DOE84] then contains detailed instructions and questionnaires for performing vunlnerability assessments. The vulnerability assessment process has four steps as follows:

1. Analysis of the General Control Environment (via questionnaire 1, that looks at management, organization structure, and personnel facts)

2. Analysis of Inherent Risk (via questionnaire 2, that looks at factors such as purpose, budget allocations/outlays, degree of centralization)

3. Preliminary Evaluation of Controls (via questionnaires 3I, 3II, 3III that look at operations, applications, and microcomputer controls respectively)

4. Summarize Results (via a summary sheet)

The evaluation process requires a judgment by the evaluator that selects a value from a possible high, medium, or low numerical value that has been assigned to each pertinent factor in each of the questionnaires. The evaluator's selected values are then summed within each questionnaire and the 'numerical values' obtained for each questionnaire are then added together to give a numerical assessment score. The summary sheet [DOE84, p. 83] assigns ranges for interpreting this assessment score as high, medium, or low vulnerability. The methodolgy is heuristic with the possible values assigned to factors being based on the experience of the developers of the methodology. In February 1985 a survey was taken by DOE among the approximately 500 recipients of this document. 62% of the 250 respondents found it useful.

The latter half of the document discusses internal control techniques for the three major control categories (management, operations, and applications). These can not only be used as a menu for control choices for new or modified systems, but can also be used as a comparison base when performing a vulnerability assessment.

DOE's draft document on performing internal control reviews [DOE83] gives guidance on both planning and performing the review. Important planning considerations are: staffing, assembling permanent review material, understanding the organization, documenting the review, and assigning responsibilities. The major activities in performing the review consist of:

1. Identifying the event cycles [steps to get something done]
2. Analyzing the general control environment
3. Documenting the event cycle
4. Establishing the control objectives
5. Identifying the control techniques
6. Testing the internal controls
7. Evaluating the control system
8. Reporting the results

These have been divided into fifteen steps and have five exhibits for assisting in the documentation. Evaluations are again judgemental (adequate, inadequate) but no scoring scheme is used. The event cycle component draws heavily on a work by Arthur Andersen & Co. entitled "Guide for Studying and Evaluating Internal Controls in the Federal Government."

4.1.6.12 Formal Verification - A technology assessment of ways to measure computer security would not be complete without mention of the current work being done in formal design and program verification. While still largely an area of research (despite many claims to the contrary), formal verification has promise if it can be empirically shown to be cost effective.

The formal verification community has long stressed that "the certification of software systems has not . . . been firmly rooted in sound engineering techniques.... the available techniques . . . rely heavily on ad hoc examination and approval of software" [BEL74]. Formal verification methodologies have the theoretical potential to contribute substantially to the improved management of the software (and perhaps hardware) security evaluation problem. This is forcefully illustrated by a conclusion arising out of the USC/ISI work on protection flaw categorization.

> "During the research effort one thing that became evident was the role of program verification techniques in detecting operating system security vulnerabilities. It is hard to see how true, definitive statements about the security afforded by an operating system can ever be made until program verification techniques have been applied" [BIS78, p. 16].

There are two major forms of formal verification: design verification and program (i.e. implementation) verification. The total process involves (1) defining a set of "correctness" criteria, (2) mathematically proving that the design specifications abide by the criteria, and (3) proving that the program obeys its specifications. Note that this is a general process which is not necessarily intended for security purposes. Its primary purpose is to improve the quality of specifications by imposing more precision and rigor on their development. Use for security purposes involves including security properties (or axioms) as correctness criteria. The computer security requirements for data accuracy and reliability as well as system accuracy and timeliness cause some security properties to be identical with the "correctness" criteria of the formal verification process.

Today's formal design specifications are usually state descriptions [24]; therefore proving that the specifications abide by correctness criteria would involve showing that:

- o No other states can occur
- o The set of transitions between states is complete
- o Certain security properties hold.

Lauer presents a description of program verification [LAU76, p. 19].

> "Formal verification is the application, in a rigid and algorithmic fashion, of mathematical and logical principles to the problem of certifying the computer programs are correct and consistent, and comply with their specifications. When we say that we have formally verified a program, we mean that we have demonstrated by a rigorous, precise argument that the representation of that program conforms completely to the representation of its specifications. The word representation is important here because the art of formal verification deals with the <u>forms</u> of program and specifications. A set of transformations is defined which turns the sequences of symbols representing a program and its specifications into well-formed formulas of mathematical logic. The underlying theory of formal verification has, as its principal result, that a program satisfies its specifications if and only if the corresponding well-formed formulas can be proved to be theorems. By constructing such proofs - based on formal, precisely stated axioms which characterize the programmer's knowledge about this problem, system, and universe - we provide a positive, unambiguous certification of that program with respect to those specifications."

---

[24]An exception is the USC/ISI AFFIRM methodology which uses algebraic specifications [GER80].

Much progress has been made over the last few years in formal verification. A number of prominent methodologies exist including:

- o Gypsy, University of Texas at Austin
- o Hierarchical Development Methodology (HDM), SRI International
- o Formal Development Methodology (FDM), System Development Corporation
- o Stanford Verifier, Stanford University.

As noted in a survey [CHE80, p.1] these methodologies "should be considered experimental . . . . They are undergoing continuous evolution . . . none should be considered final products". Problems remaining include such issues as the following [MIN80, p. 7]:

> "Specification languages do not permit representation of all aspects of the system that are relevant to security. Some of the aspects that are inadequately represented or left out of current specifications are: execution times, concurrency and asynchronous I/O; interrupts; parameter passing mechanisms; scheduling; and process creation and deletion."

The program chairman of a workshop in formal verification concluded that there was "immediate evidence of progress in design proofs" and that program proofs were "beginning to be feasible for realistically sized program units" [NEM80, p. 7]. He also concluded that "verification will not have a real impact until it is applicable to real systems, real programming languages, and real requirements to be proved". There is work by Applebaum and Keaton-Williams at the Mitre Corporation in Bedford, Massachussetts on the design of a practical verification system (PVS) [APP83].

Formal verification techniques have been used for security in several development efforts in the DoD community (e.g. the Kernelized Secure Operating System (KSOS) and the Automatic Digital Network (AUTODIN II)) [CCT78]. Formal verification has resulted in the detection of security flaws and clearly improves security defenses. The main question which must be resolved is whether the additional security justifies the added cost. There are many issues which are relevant here:

1. Costs will fall as techniques are refined; it is possible that evidence will appear in the near future showing partial verification (e.g. use of a formal specification without a proof, or use of a proof at the design but not implementation level) to be cost effective for the structural integrity and quality it provides. Of course the "Hawthorne Effect" from industrial psychology [25] might serve as a warning against placing too much credence on early results. Evidence will be more reliable when it is not accumulated under the gaze of

the verification research community.

2. Formal specifications are very difficult to prepare. Some or even many designers and programmers may not be capable of working under the rigors imposed by formality. This might be an advantage, since it could be argued that such people are not ideal for developing security-critical software.

3. Disagreement and confusion exist as to the benefits and limitations of formal verification [26]. There is a consensus that it will not supplant the need for normal functional testing or security penetration testing. Gerhart and Yelowitz, in noting the "obvious" applicability of proofs to certification , stress their limitations by citing several "proofs" for which counterexamples were subsequently found [GER76]. They conclude that formalism is "one, but not the only, or necessarily best tool for verifying programs" and state that "experience with both testing and mathematical reasoning should convince us that neither type of evidence is sufficient and that both types are necessary" [GER76, p. 206].

4. Formal techniques may not be appropriate for systems which change and evolve at a high rate. It is also highly unlikely that formal techniques could be introduced after a system was developed in order to facilitate security evaluation.

Formal verification has promise as a security evaluation methodology. It has the potential to basically alter the software (or hardware) development process. It also has the potential to be shown too cumbersome and costly for more than very limited use. Until its best use is determined, formal verification remains an intriguing security evaluation and certification approach which must be considered in future evaluation work.

---

[25] In the Hawthorne studies [ROE39], factory workers produced more and more, despite positive and negative experimental manipulation of their working environment. It was concluded that the key factor in increasing worker productivity was the attention and interest of the researchers.

[26] The most widely referenced critique of verification is [DEM79, pp. 271-280]. Community response to the paper can be found in [ACM79, pp. 621-630].

4.2 RISK ASSESSMENT METHODOLOGIES

Risk assessment is a method for estimating the anticipated or expected loss from some adverse event. Already used in risk management investment decisions and in insurance risk calculations, when applied to ADP systems risk assessment is a systematic quantitative and qualitative procedure for estimating the level and allocation of security safeguards.

Risk assessment has been used in areas other than computer systems, notably the assessment of insurance risks, assessment of risks of new technologies (such as nuclear power plants), etc. The area, however, is new and still developing. Even at this early date there are quite a few risk assessment methodologies used for computer systems.

4.2.1 Individual Methodologies

This section describes the following methodologies:

- FIPS PUB 65 [FIP65]
- Air Force Risk Analysis Management Program (AFRAMP) [AFRAMP]
- Department of Agriculture [DOA77]
- SDC Navy Risk Assessment Methodology [SDC79]
- Risk Analysis and Management Program (IST/RAMP) [IST79]
- Relative Impact Measure (RIM) of Vulnerability [NIE80]
- Fuzzy Risk Analysis [HOF80]
- Security Assessment Questionnaire [IBM80] [IBM85]

Much of the analysis presented in the first four summaries is adapted from [MOR80].

At the present time a number of groups are developing and/or using automated risk analysis packages on microcomputers. A discussion of these is beyond the scope of the present document. Some of the ones the Editor is aware of at this time are: RASYS for an IBM PC, which was based on algorithms of John Carroll and is beng used by Nander Brown at the Federal Home Loan Mortgage Corporation in Washington, D.C.; an extension of Visicalc by Bound and Ruth [BOU83]; a package called LAVA (Los Alamos Vulnerability Assessment) developed by Suzanne Smith at Los Alamos Laboratory in Albequerque, N.M. for a COMPAQ microcomputer; a program called RISKCALC being developed by Dr. Lance Hoffman of George Washington University, at first using an APPLE IIe and now on an IBM PC; a package called RISCALC being developed for an IBM PC at EDP Audit Controls, Inc. in Oakland, California; a package called Audit Risk by RTS Software Systems, Inc. in New York City; an automated risk profile package called RiskPac developed by Peter Browne of Profile Analysis Corporation in Ridgefield, Conn.; a risk analysis system RISKA that runs on any microcomputer system

operating under MSDOS 2.10 with 256K memory [RAM] and 10 megabytes of storage and that was developed by Shig Tokubo of Expert Systems Software, Inc. in Danville, California; and a RISK ANALYSIS MACHINE written for a Kaypro 10 by Don Colner of Basic Data Systems, Inc. in Rockville, Maryland.

4.2.1.1 FIPS PUB 65 [FIP65] - This document is a follow-on to FIPS PUB 31, "Guidelines for ADP Physical Security and Risk Management", which was for some time the principal reference for Federal ADP Managers on risk management. Developed by the National Bureau of Standards and based upon work done by Robert Courtney of IBM and others, it evolved over the period of approximately 1972- 1979. One of its primary objectives is to inform the Federal ADP Manager how to do a risk assessment.

The methodology calls for three basic activities: a preliminary examination, the risk analysis, and the selection of cost-effective countermeasures. Risk analysis itself requires estimation of two quantities: (a) frequency of occurrence (annualized) of a threat and (b) impact when a threat materializes and affects some asset. Multiplication of these factors results in the annual loss which can be expected from that threat. Summing over all threat-asset pairs provides the annual loss expectancy (ALE). A risk analysis worksheet is provided which allows entering raw data for the computations of ALE. For each system asset, it is necessary to a) identify each threat which can impact the asset; b) estimate the expected frequency of occurrence of the threats in each impact category; c) estimate the impact in dollars of each threat in each of the four categories (modification of data, destruction of data, confidentiality of data, and denial of service); d) multiply these ratings to obtain the ALE for each asset-threat pair; and e) sum over all such asset-threat pairs to obtain the system-wide ALE.

The fundamental operand here is the asset. Data files, equipment, software, personnel, communications, negotiable output, etc. are all assets. There is a heavy emphasis on procedural and physical safeguards. An appendix containing system vulnerabilities is provided.

As a high level statement of the purposes and methods of a risk assessment, FIPS PUB 65 does very well. Its major strength is that it produces dollar figures which can be, in the best of circumstances, reasonable estimates for annual loss expectancies which can then be dealt with by managers. Many people using this methodology have found that the use of an orders of magnitude matrix for dollar loss and frequency of occurrence is an additional strength.

Its major disadvantages are that it requires a highly skilled group of people working together to produce these estimates and that dollar values as a metric may not be universally applicable. Some people consider a major weakness to be that it does not provide

detailed guidance on how to actually perform the risk analysis. It relies in large measure on the expertise and judgement of the members of the risk analysis team. However, the NBS view has been that this level of detail is unique to each organization and therefore cannot be described universally.

4.2.1.2 Air Force Risk Analysis Management Program (AFRAMP) [AFRAMP] - This document, until about 1980 comprised a major part of the Air Force ADP Security Program. The methodology has now been shelved and is being replaced by a set of questionnairs. The AFRAMP will be described here since it represents a major risk analysis effort that was in process when this Technology Assessment effort was being performed.

The AFRAMP consists of three volumes. The first is a Risk Assessment to be applied to Air Force ADP facilities. The second and third, interestingly, contain the procedures which must be gone through to certify application software (Vol. II) or systems (Vol. III) as appropriate for processing of sensitive data. The distinction between the contents of Volume I, on the one hand, and Volumes II and III, on the other, is: Volume I evaluates an in-place, operational facility for risks, quantifies the expected loss, and recommends cost-effective countermeasures (i.e., conducts a risk assessment); Volumes II and III provide extensive, detailed guidance on how to certify application software and tactical or command/control systems when classified materials will be involved. This distinction points out the differences between the concerns of general ADP managers and the concerns of the military and intelligence communities. In one sense, the difference is one of degree, since in both cases threats and vulnerabilities are analyzed and assessed. However, a risk assessment for ADP managers usually proceeds on a much higher level (i.e., with less detail) than the detailed, technical testing and review which is often required of systems which will process military and intelligence data.

The AFRAMP is a highly structured sequence of procedures set forth in complete detail. Whenever possible, explicit activities are described with painstaking specificity, including the exact procedures to be followed and the output which will be produced by the activity. As with FIPS PUB 65, the major summary statistic is the ALE, although other measures of the level of risk are computed. As usual, the ALE is taken to be the frequency of occurrence of a threat multiplied by the value of the affected asset. Summing over all threat-asset pairs results in a system-wide ALE. The AFRAMP methodology follows an eight stage approach which involves inventory of assets, threat identification and evaluation, computation of ALE when no ADP safeguards are in place, inventory and evaluation of existing security measures, computation of ALE when safeguards are in place, measures of the level of security, selection of countermeasures, and action and reporting. The most siginificant measure of the level of security is the annual percentage of loss, which is just the ALE expressed as a

percentage of the assets at the facility.

AFRAMP has a number of strengths and weaknesses. The strengths include: a highly structured, methodical approach; extensive guidance on evaluation of assets and estimation of threat frequencies and magnitudes; a carefully conceived mechanism for selection and evaluation of countermeasures; unambiguous assignment of responsibility; and review and approval procedures at several points in the process. The potential weaknesses include: a very high level of effort required to carry out the AFRAMP and failure to allow an asset to be evaluated in more than one mode of impact (i.e., inability to distinguish destruction, disclosure, denial of service, etc.). There is little if any actual experience using AFRAMP.

4.2.1.3 Department Of Agriculture [DOA77] - The U. S. Department of Agriculture has produced a security handbook intended for use 1) as a means of assessing the current security position; 2) in raising overall security awareness; and 3) as a management tool for cost-effective allocation of resources. It presents a methodology very similar to that of FIPS PUB 65, except that orders of magnitude estimates are not permitted. Only two classes of risk -- major and minor -- are used. The handbook consists of 20 pages of which a number are used to present a carefully worked example. The principal categorizing entity is threat rather than asset. Users are involved and required to identify critical system assets and services, assess the sensitivity of data files under their control, specify what additional security measures if any are required, and estimate the impact associated with the occurrence of major and minor threats. The methodology also presents an orderly approach to the selection of countermeasures.

A major strength of the methodology is the active involvement of users in the evaluation of the impact resulting from threat occurrence. As in FIPS PUB 65, the directions for implementation are sufficiently general that assessments involving widely differing levels of detail and effort could reasonably be said to fit under the same description. While this allows ADP managers flexibility in choosing the amount of time and resources to be allocated, such decisions will have a great impact on the validity and utility of the results of the assessment. The major weakness in the USDA methodology is lack of guidance in arriving at a suitable list of threats and in estimating their frequency and impact. A team approach is recommended, but the size and content of the team is left open. USDA has shelved this handbook and now refers its agencies to [FIP65].

4.2.1.4 SDC Navy Risk Assessment Methodology [SDC79] - The SDC risk assessment methodology developed in 1979 for the Navy consists of six phases. In the initial three phases, the threats, vulnerabilities, and assets of the system are identified and evaluated. Checklists for

these items are given. Generic lists of threats and vulnerabilities are supplied although others can be added as appropriate.

Assets are inventoried and are then evaluated in each of the four modes of impact: unauthorized destruction, disclosure, modification, and denial of service. In the estimates, orders of magnitude are used. Vulnerabilities are rated using qualitative verbal descriptions such as "very low" or "medium". These verbal descriptions are then transformed into numbers based on mathematical tables given in the methodology. Assets can be evaluated using either dollars or (since the impact of certain threats is very difficult to evaluate in dollars) a qualitative non-dollar value technique. After the initial three phases, threats are matched against vulnerabilities to obtain plausible attacks in the areas for each of the four modes of impact. A set of forms is supplied to help in this process. A successful attack frequency is computed (again using underlying mathematical tables). The frequency of attacks is multiplied by the impact (in this case asset value) to yield the ALE. The sixth stage consists of selection of countermeasures.

This technique also requests a precision rating when threat frequencies and asset values are estimated. This rating indicates how accurate the evaluator feels his or her estimate to be and attempts to take into account evaluator error.

A key distinguishing feature is the explicit treatment of vulnerabilities. This tends to make the analysis more situation-specific than the sole use of more generic threat-asset pairs. It also results in a more scenario-oriented analysis. Other distinguishing features are the specification of a precision rating by the evaluator, the eight forms required to carry out the analysis, and the possibility of using non-dollar valued assets. These features can also be seen as disadvantages, particularly the necessity for having a potentially inexperienced person using eight different evaluation forms, and the underlying uncertainty involved with computing dollar figures from non-dollar valued assets. Nonetheless, it is an interesting step beyond the earlier methods and recognizes some of the (legitimate) concerns about confidence and imprecision of ratings.

4.2.1.5 Risk Analysis And Management Program (IST/RAMP) [[IST79] - IST/RAMP is a proprietary program owned by International Security Technology, Inc. (IST) of New York City and developed in 1976 by Robert Jacobson, President of IST. For a few years in the early 1980's it was marketed by PANSOPHIC and called PANRISK. Most recently a front end enhancement has been developed for it by Duncan and Associates to make data entry and output more user friendly. IST/RAMP is an automated method for obtaining quantitative estimates of expected losses caused by threats materializing at data processing facilities that handle multiple applications. It is meant to be used by a security analyst of some kind.

IST/RAMP automates the computations necessary to do certain types of risk analyses, notably those related to a central data processing shop; some teleprocessing threats can be considered as well, but the non-central site teleprocessing facilities do not appear to be handled integrally with the central site as part of the same system.

IST/RAMP currently operates on the IBM DOS and OS systems whereas the original IST/RAMP operated on the General Electric commercial MARK-III time-shared computer system. This software computes the expected loss per year for three asset types: applications, master files, and rooms plus contents. It does this by refinements of the well-known relationship

Expected Loss = Threat Probability x Loss Potential.

These refinements include the use of an average threat occurrence rate rather than a threat probability and an appropriate heuristic interaction function that modifies the loss potential for each category of threat. The expected loss due to a materialization of each threat is then calculated for each asset, taking into account the interaction function.

Threats are categorized by types of loss that they produce, i.e. delay (denial of service), fraud via EDP, information disclosure, physical damage, and physical theft. The interaction function for delay loss takes into account the time distribution of delays as well as a slack time adjustment (time during which system does not operate and therefore cannot be delayed); the interaction function for fraud takes into account vulnerability factors; etc. Table 4-13 shows which loss category applies to which asset type and what the related interaction function considers.

The software allows one to sum all expected losses due to a given threat or associated with a given application. The package has its own threat data base which the user can modify via programs. The user puts in his own "rooms" and delay loss data.

Table 4-13. Loss Category, Asset Type, Interaction Function Relation

Loss Category	Asset Type	Interaction Function
Delays	Applications	Delay distribution and slack time adjustment
Fraud via EDP	Applications	Vulnerability function
Info. disclosure	Master files	Vulnerability function
Physical damage	Rooms and contents	Threat effect factors
Physical theft	Rooms and contents	Threat effect factors

One interesting feature of the program is that it can help optimize back-up plans from the delay loss viewpoint. A second interesting feature is its ability to estimate single occurrence losses that may occur very infrequently but have a very severe effect on an organization. Further interesting features are: iterations of the program allow for optimizing the selection of safeguards; the information that this program handles can also assist in feasibility studies for new applications; and finally, it can assist in cost studies for hardware configurations with variable redundancy and various delay losses due to hardware failure.

IST/RAMP is completely numeric; it presents the expected losses arranged in descending order. Even if some data is missing, however, it produces numerical answers. Thus, there is a potential problem when there is very little actual data to go on, i.e. the results in these cases could look useful but be meaningless. The program needs to be operated by a security evaluator type individual in order to enter data and attach appropriate significance to the results.

IST/RAMP produces several reports; two of the most important are highlighted here: expected damage and delay loss.

Expected Damage. Expected damage is computed based on inputs from the threats and rooms data bases supplied by the user. This information is presented by room or by threat, as shown in Figure 4-11.

Delay Loss. Losses arising from delays in processing are computed and presented by threat or by application; the applications are grouped into vital, critical, important, and non-urgent categories. Expected or single-occurrence losses can be presented. An example of such a report is shown in Figure 4-12.

MARCH 10,1979
ROOMDB03(9) AND USTHRT02(31)

NUM.	ROOM NAME	EXP.DAM $S	PCTN.	CUMM.
1A100	COMPUTER ROOM	143241.	74.82	74.82
1B206	AIR CONDITIONING MACHINERY R	46458.	24.27	99.09
1A106	U.P.S. UTILITY ROOM	731.	0.38	99.47
2A400	APP. PROGRAMMERS OFFICE	389.	0.20	99.68
1A105	TAPE LIBRARY	259.	0.14	99.81
1A104	FORMS STORAGE ROOM	248.	0.13	99.94
2A500	TECHNICAL LIBRAY	70.	0.04	99.98
2A450	SYS. PROGRAMMERS OFFICE	42.	0.02	100.00
2A350	D.P. MANAGER'S OFFICE	0.	0.00	100.00
	TOTALS	191438.		

Expected physical damage for each of the rooms in an illustrative data processing facility.

MARCH 10,1979
ROOMDB03(9) AND USTHRT02(31)

NUM.	THREAT NAME	EXP,DAM $S	PCNT.	CUMM.
20	MINOR COMPUTER ROOM FIRE	125000.	65.30	65.30
15	PLUMBING FLOODING,ROOF LEAK	41356.	21.60	86.90
16	GAS EXPLOSION IN BLDG	10858.	5.67	92.57
21	COMPUTER ROOM BURN-OUT FIRE	6244.	3.26	95.83
23	MAJOR FIRE ELSEWHERE IN BLDG	4561.	2.38	98.21
22	D.P. AREA BURN-OUT FIRE	2929.	1.53	99.74
24	BUILDING BURN-OUT FIRE	162.	0.08	99.83
33	ANTI-COMPUTER DEMONSTRATION	149.	0.08	99.91
43	MODERATE EARTHQUAKE MM = 8	85.	0.04	99.95
44	MAJOR EARTHQUAKE MM = 9	53.	0.03	99.98
34	D.P. OPERATION STAFF STRIKE	41.	0.02	100.00
30	BUILDING BOMBING	0.	0.00	100.00
37	D.P. AREA BOMBING	0.	0.00	100.00
	TOTALS	191438.		

The expected damage with respect to each of the threats for the data processing facility.

Figure 4-11. Illustrative RAMP Reports [IST79, p. 20].

EXPECTED ANNUALIZED LOSS FOR EACH APPLICATION

APPLICATION NO.	APPLICATION NAME	EXPECTED ANN. LOSS (000'S OMITTED)		PERCENT OF TOTAL
	VITAL APPLICATIONS			
10	LINE CREW DISPATCHING	$ 679	$ 679	28.656
8	ENERGY DISPATCHING	$ 481	$ 1,160	20.291
	CRITICAL APPLICATONS			
14	CASH MANAGEMENT SYSTEM	$ 384	$ 1,545	16.233
	IMPORTANT APPLICATIONS			
21	NUCLEAR FUEL CONTROL	$ 321	$ 1,866	13.544
15	SPARE PARTS MANAGEMENT	$ 269	$ 2,136	11.346
	NON-URGENT APPLICATIONS			
2	CUSTOMER ACCOUNT STATUS	$ 178	$ 2,315	7.548
3	CUSTOMER STATEMENT CALC+PRT	$ 33	$ 2,348	1.417
9	GENERATION ENG. CALC'S	$ 14	$ 2,362	0.603
1	CUSTOMER RECEIPT POSTING	$ 7	$ 2,370	0.307
6	GENERAL LEDGER	$ 1	$ 2,371	0.043

Figure 4-12. Typical RAMP Report [IST79, p. 21].

4.2.1.6 Relative Impact Measure (RIM) Of Vulnerability [NIE80] - This technique is still in its embryonic stages. Developed in the late 1970s by Norman Nielsen and Brian Ruder and others at SRI International, this methodology measures the relative impact on an organization of vulnerabilities of its computer system integrity. It presents relative measures between two competing systems or configurations. The developers believe that using relative measures rather than absolute measures has more advantages than disadvantages, in particular these: (1) obtaining relative values is much easier from a human engineering point of view (and possibly more accurate also); (2) the approach provides credibility since the relative numbers are more meaningful to the persons performing the evaluation and to management. The disadvantages, of course, include the fact that there are no bottom-line figures presented to management. However, the methodology is easy and inexpensive to apply and sometimes more suitable in practice than a monetarily-oriented technique, since users are often unable to make estimates of expected violation frequencies.

The methodology has four basic steps: perpetrator analysis, target assessment, flaw identification and analysis, and relative impact measure (RIM) calculation. RIM is designed for use with computer facilities; it merges the aforementioned elements together to produce a metric.

The first step, perpetrator analysis, initially determines the number of people in each of a series of categories relating to individuals both inside and outside the organization. Then, the people in these categories are distributed among (nine) perpetrator classes. Next, the relative severity of impact caused by members of each perpetrator class is computed. Finally, a procedure is used to provide a likelihood (for each class) of "attack".

The second step, target assessment, initially assigns individual target elements (e.g., files, equipment) to one of nine target classes. Then a procedure is used to assess the relative impact of a successful attack upon an element of each target class. Finally, using information obtained from the perpetrator analysis, the relative attractiveness of each target class as a potential target for each perpetrator class is computed.

The third step, flaw identification and analysis, first attempts to identify the exploitable flaws in the system and then develops "flaw-probability estimates" linking each perpetrator class with each target class. Finally, the measures provided from the previous three steps are combined, in essence using a sum of products.

The RIM, like fuzzy risk analysis (see below), expressly acknowledges the subjective nature of some evaluations that are required to calculate the RIM value.

4.2.1.7 Fuzzy Risk Analysis [HOF80] - This is a method of risk analysis developed in 1980 by Hoffman and Neitzel. Also still in the research stage, this method accepts as input a system described as a collection of subsystems (in a tree structure) along with estimates of severity of loss and likelihood of fault for the lowest level elements, and weights for the higher level elements. An example of the input data for a hypothetical computer center is shown in Figure 4-13.

The rationale behind this system is that since so many risk decisions and evaluations are made based on very little data, and since so many risk estimates are subject to large error, it is better to use "fuzzy" linguistic terms rather than numbers and to report estimates in linguistic terms to avoid giving the user an unwarranted confidence in the results of the risk analysis. This system also accepts as input confidence indicators related to each input estimate, and attempts to weight the estimates accordingly.

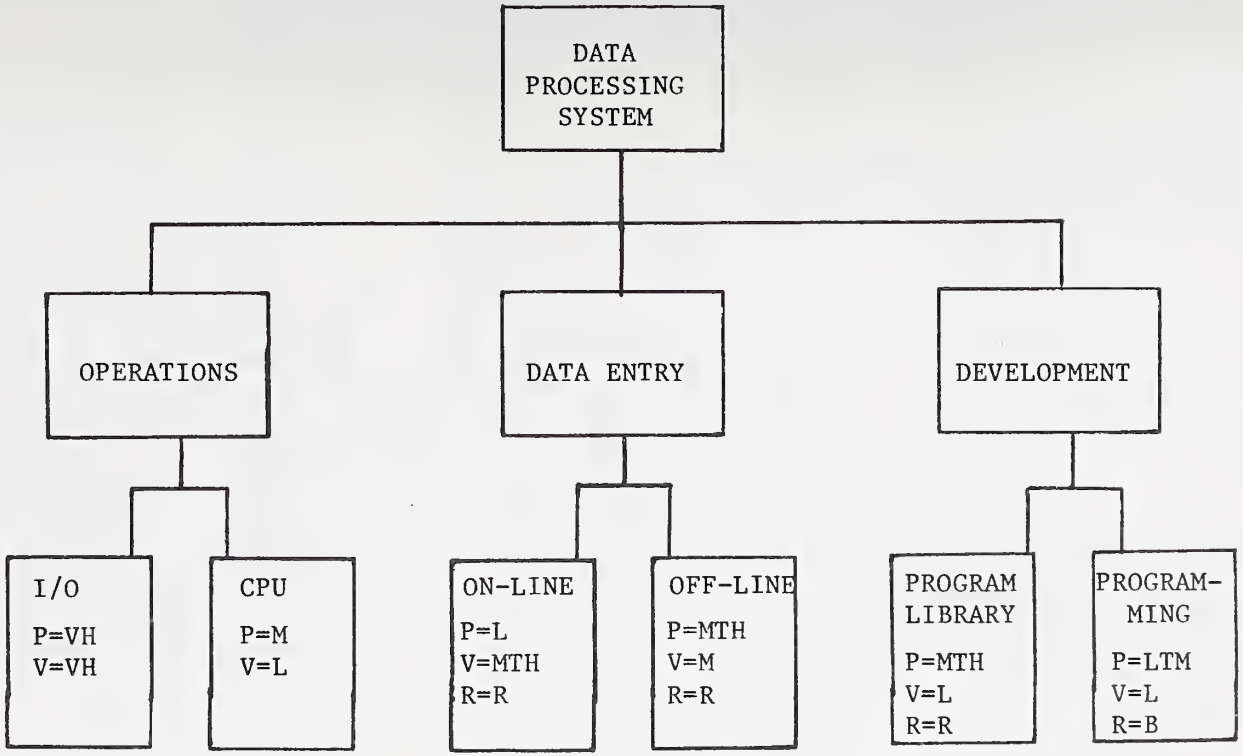
As shown in Figure 4-13, fuzzy risk analysis can be applied to computer systems. In addition, unlike many other systems described above, it can be applied to any system for which one is attempting to assess risk. In fact, [HOF80] presents an example using an automobile system and subsystems rather than a computer.

Given this information, a risk indicator for the system (and for each subsystem) is computed using mathematical methods grounded in fuzzy set theory [ZAD75]. Figure 4-14 shows computed risk indicators for the subsystems and system of Figure 4-13.

More than any of the other techniques above, fuzzy risk analysis allows establishment of hierarchical levels of detail to permit the actual estimates of severity of loss or likelihood of threat to be made at an appropriate level. Other approaches, while not prohibiting this, do not include it explicitly. On the other hand, fuzzy risk analysis is still a research tool; better documentation and user-friendly input protocols will be needed before it will see widespread use. It is currently not being actively worked on.

The major strengths here are that numerical estimates are not required. To the manager who wishes just an overall assessment and a quick feel for where his or her major risks are, the method seems very promising. The major weakness is that numerical estimates are not yet allowed. The ultimate idea of the researchers is to have both numeric and non-numeric estimates allowable, and to use the more appropriate in each case. Another disadvantage is that the method also does not provide a checklist. A third disadvantage is it considers risk alone; cost is currently not handled. On the other hand, it and IST/RAMP are the most simple to use to perform sensitivity analyses because they are automated.

A FORTRAN prototype for fuzzy risk analysis runs on both a microprocessor and the IBM 370. More recently, a prototype interactive version has been under development.



Bottom-level values are comprised of P (probability or likelihood), V (severity) and R (reliability or confidence). Key: L=low, M=medium, H=high, T=to, V=very; R=reasonably reliable, B=barely reliable.

Figure 4-13. Input Data to Fuzzy Risk Analysis for a Hypothetical Computer Center.

```
        FUZZY ANALYSIS RESULTS
        ----------------------
P = PROPORTION OR PROBABILITY
V = VALUE OR SEVERITY
R = RELIABILITY

        KEY    ELEMENT                P      V      R
        ---------------------------------------------

        1000000 DP SYSTEM             *      NH
        1100000 OPNS                  *M     MTFH
        1200000 DATA ENTRY            *M     FLTM
        1300000 DEVELOPMENT           *M     OL
        1110000 I/O OPNS               VH    VH
        1120000 CPU OPNS               M     L
        1210000 ON-LINE DATA ENT       L     MTH    R
        1220000 OFF-LINE DATA ENT      MTH   M      R
        1310000 PROGRAM LIBRARY        MTH   L      R
        1320000 PROGRAMMING            LTM   L      R

FUZZY ANALYSIS COMPLETED.
```

COMPUTED RISK INDICATORS

NH=not high
MTFH=medium to fairly high
FLTM=fairly low to medium
OL=more or less low

Figure 4-14. Example of Fuzzy Risk Analysis Output

4.2.1.8 Security Assessment Questionnaire [IBM80] [IBM85] - The questionnaire contains fourteen categories that are divided into three key security areas: physical security, controls and procedures, and contingency planning. The categories include such areas as fire, operational controls, and back-up. Each question requires a "yes" or "no" answer with the "no" identifying risks or the need for safeguards. At the end of each category is a space for rating the entire category, as follows:

A - extremely low risk - might relax controls
B - necessary risk - no action
C - acceptable risk - might perform corrective action
D - high risk - need corrective action

At the end of each category there are also references to appropriate publications so that the user can acquire more skills.

The advantage of this questionnaire is that it is relatively brief - six 8x10 pages - and yet covers a great many of the areas that are of prime concern in the security arena. It gives a manager or auditor a suitable framework for a quick assessment of the security status of a facility with sensitive applications.

The big problem is that there is no guidance on how to arrive at the A,B,C,D rating for each category. That is left to the judgement of the assessor and, consequently, the results are very dependent on the skills of the people who use the questionnaire.

There is a revised version of this questionnaire [IBM85]. The major difference is that it has added a fourth key security area, namely, policy/organization.

4.2.2 Evaluation Of Risk Assessment Methodologies

This section evaluates the risk assessment methodologies against a set of evaluation criteria. Eight evaluation criteria are used.

1. Data Completeness. The central issue addressed here is whether mechanisms are provided by the method for the construction of lists of objects, events, situations, relationships, and so forth. This includes the question of whether features are available which help to insure the completeness of these lists which are used as input data to the risk analysis of a particular system. Many risk analysis methodologies such as the SDC RAM and IST/RAMP provide such checklists (of threats, countermeasures, assets, etc.). These are often very helpful in getting a risk analysis started. Generally, the provision of such a checklist is a plus.

2. Level of Effort (cost and time required). Some of the methods such as AFRAMP, SDC RAM, IST/RAMP, and FIPS PUB 65 require detailed procedures to be carried out, either taking a great amount of time, or a very high level of expertise among those who will conduct the assessment, or both. Others such as IST/RAMP, RIM, and Fuzzy Risk Analysis do not. Many of the methodologies are vague on the level of expertise and the total amount of time required. Many of them do not allow partitioning of the system into subsystems easily. All of these affect the level of effort required.

3. Ease of Use in Performing the Assessment. Some of the methodologies such as SDC RAM, IST/RAMP, and AFRAMP provide forms and detailed instructions for evaluators and people who are producing the assessment. Others have good human-engineered computer procedures for extracting input data. Others (e.g. FIPS PUB 65, Fuzzy Risk Analysis) do not fare so well in this regard. Clearly the ease of use and training material provided can be important factors.

4. Ease of Use in Interpreting the Assessment. The results produced and the ease of their use are also important to managers. Documentation provided with these results is important as well. Methodologies faring well here include FIPS PUB 65, IST/RAMP, and Fuzzy Risk Analysis.

5. Algorithm Completeness. An algorithm should measure risk to all modes--denial of service, unauthorized disclosure, destruction, or modification. It should not handle only one of these problems. The algorithm should explicitly handle both independent threats and interdependent threats. No method adequately handles interdependent risks. Some method should be provided to handle estimation error. Numerical estimates may be very inexact; in this case, the methodology should aid in showing the accuracy and reasonableness of these estimates. In some cases, variances should be explicitly specified and the algorithm should factor these variances, especially when large, into the computations.

6. Use of Available Data. The systems should use any available information on risk (earthquake frequency, etc.) and should provide such information when it is available. FIPS PUB 65, its antecedent FIPS PUB 31, and IST/RAMP do well here.

7. Ability to Easily Perform Sensitivity Analysis. It should not require a great expenditure of time or money to perform a sensitivity analysis. Changing one or two variables and running the risk analysis again to ask "what if?" should not entail a lot of work. Some systems (especially the automated ones such as IST/RAMP and Fuzzy Risk Analysis) are very simple to use in this regard while others require a considerable effort.

8. Interface with Decision Analysis. An explicit tie-in to decision procedures for management is desirable. Unfortunately, most methodologies do not provide this at this time. AFRAMP is one that does.

4.2.3 Generic Problems With Existing Risk Assessment Methodologies

There are a number of generic problems with existing risk assessment methodologies. These problems are the same for both computer risk assessment and non-computer risk assessment. Research is embryonic in many of these areas. Thus, we do not have a solution to all of these problems. However, they are important and should be recognized by anyone who is attempting to perform a comparative evaluation of the methodologies. Some problems (as enumerated in [HOH80]) are these:

1. Incomplete Knowledge of Extent or Likelihood of Risks. An example here is the effect of fossil fuel burning on climate. We just don't know the results of this at this time; we don't know the likelihood of risks. A comparable computer example is the effect of networks on the security of data. The lack of experience in the technology for connecting mainframes to microcomputers makes it impossible to accurately estimate the risks here.

2. Foregoing Clear Benefits for Ambiguous Risks. Here, a good security example is that of operating system penetration. We know that we become vulnerable when we allow users to program on computers which are simultaneously used to support business operation. However, we get very clear benefits in terms of cost-effectiveness supporting both functions at the same time. The risks, on the other hand, are not so clear. The question then becomes this: Do we really wish to forego the clear benefits of supporting both development and operation on one machine for the ambiguous risks associated with operating system penetration?[27]

3. Limited Capacity to React. An example here is that there are 2,400 substances which may be causing cancer in the work place. How do we determine which of these are the most carcinogenic? Where do we start to control these substances?

---

[27] The most common answer to this question is well stated by Courtney [COU74, p.3]: ". . . There is a very natural, human tendency to consider things which might happen but which have never been known to happen . . . . We must reserve our concerns for those things which happen with a sufficiently high probability to justify corrective measures, including, where appropriate, recovery rather than avoidance."

Which ones? Given that we do not have resources to attack all of these, or that some of our solutions may be worse than the problems, a case can be made for doing nothing. In the computer field, the opportunities for security breaches in a system are very numerous. The question of where to place safeguards and how much to spend, regardless of the risk analysis method used, still requires a good deal of judgement and may not always result in sufficient security improvement relative to cost.

4. Perception and Measurement Problems. Perceived risk varies over a factor of less than 30 while actual risk as measured by experts varies over a factor of 1 million. As one example, most people judge risks from auto accidents about the same as risks from nuclear power. Granted that we have much, much more information on auto accidents, the "experts" (at least) believe that ordinary people are wrong. (This of course avoids the possibility that the experts have built in biases, which we will not address here.) Data is often absent, in which case one can extrapolate, transfer experience, or use fault trees or event trees to attempt to establish a reliable base. On the other hand, this is exactly what was done in the Rasmussen Report dealing with nuclear reactors; it has been generally thought a good effort, but nevertheless wanting (in the light of Three Mile Island). There are also problems with uncertainties at low probabilities and limited data availability. All of these considerations are valid in the computer field as well.

5. Value Trade-Off. Here, a good example is nuclear waste storage. Do we want the relatively cheap energy from nuclear power, given the cost of having to store the nuclear waste somewhere? Where is somewhere? It may be fine as long as it is not your backyard. This raises various equity and other problems which are not addressed by most risk assessments (and perhaps should not be; rather, they should be assessed by cost-benefit or decision analyses which follow the risk analysis). In the computer field, a parallel discussion could be conducted for contingency planning. Too little planning could be fatal to the organization while too much planning could generate excessive costs for the value received.

6. Institutional Weakness. An example here is where the Department of Transportation widens the shoulders of interstate highways at a cost of $6,000,000 per life saved. There are cheaper ways to save more lives, but DOT has in the past done this rather than something else (such as fighting the political battles for (for example) mandatory seat belts, which would save many more lives per dollar spent). It is only recently that steps in this direction have been taken in this direction by considering airbag devices. An example in the computer field is the maintaining of obsolete computer

systems rather than making the decision to redesign systems and update the technology. The decision to redesign usually involves an uphill battle against the status quo and is therefore undertaken less frequently than advisable.

7. Creation of New Hazards. It could be argued that compulsory driver education accelerates the entry of high-risk drivers onto the road. In like manner, the introduction and discussion of various safeguards may educate more users on the vulnerabilities and flaws of computer systems and tempt more people to exploit them.

These are some of the generic problems with existing risk assessment methodologies. These issues should always be considered before we rush to judgment to obtain a method for assessing competing risk assessment methodologies. We still have a lot to learn about the topic, although it has progressed sufficiently to be a useful tool in the right circumstances.

CHAPTER 5

SUMMARY OF THE STATE OF THE ART

This chapter summarizes the findings obtained from examining the security evaluation and risk assessment methodologies described in Chapter 4. It begins by identifying similarities and differences among the methodologies. Distinctions are drawn between EDP audits and security evaluations (categorized together above, due to their overlapping nature, as simply "security evaluation" methodologies). Differences between audits, security evaluations, and risk assessments are analyzed and a comparison of the generic analysis approaches (e.g. matrix, transaction flow) is presented. The general evaluation issues of quantification, uncertainty and bias, and integration with the decision process are discussed. Finally, some general conclusions are presented.

5.1 SIMILARITIES AND DIFFERENCES

The methodologies described in Chapter 4 all have several characteristics in common. Most significant is that the general underlying structure is fairly constant. This is evident more from the implicit intent of the methodologies than from their explicit breakdowns of tasks or evaluation steps. While there are a few variations[28], the basic structure of the security evaluation process in the audit, security, and risk assessment communities is fairly similar[29].

A second common feature is that all of the methodologies, in their attempt to simplify the evaluation process, have neglected to

---

[28] The Canadian Institute of Chartered Accountants' approach [CIC75] includes an interesting variation; detailed consideration of vulnerabilities and exposures occurs only after the control evaluation, as a step in determining compensating audit procedures. While unusual in the audit community, this is similar to the practice in many risk assessment methodologies of addressing countermeasures (i.e. controls) only after the initial evaluation has produced a loss estimate.

mention relevant functions, issues, and interrelationships. As a result, all have in a sense contributed to a misunderstanding of the evaluation process. This is important for it is through an awareness of the actual complexity involved that the difficulties of measurement become most apparent. In simplifying the process (i.e. modelling it) for methodological purposes, one must make many simplifying assumptions (e.g. about threat frequency and severity, control interrelationships). Unfortunately, there is no body of research or canonical evidence upon which to base the necessary assumptions[30].

A final common feature is that little empirical data exists on the use, success, or failure of the methodologies. There are probably several reasons for this:

- For security reasons, organizations are reluctant to disclose their organization-specific, detailed methodologies or findings.
- Organizations may not be seriously and conscientiously using the methodologies.
- Methodologies do not have built into them validation criteria by which success or failure can be measured.

All of the methodologies are different in their specific approaches. These differences derive from different purposes (e.g. audit versus risk assessment), different control groupings, and individual approaches to analysis. Alternative control groupings were discussed in Chapter 3. The impact of different purposes and analytic approaches is discussed below. Because the most significant differences in the methodologies are such large-scale philosophic ones, the discussion of differences will focus on the differences themselves, not on the methodologies. Thus, emphasis will be placed on the advantages and disadvantages of the generic approaches, not on precisely how the individual methodologies differ from each other.

---

[29] A fourth community which can perform security evaluations is that which does Verification, Validation, and Testing [VV&T]. This activity occurs when a system is tested against its security requirements. See [FIP101] and [FIP102] for more information.

[30] The implications of this are similar to the situation which exists in Forecasting (whether policy, economic, technological, etc): "The core assumptions underlying a forecast, which represent the forecaster's basic outlook on the context within which the specific forecasted trend develops, are the major determinants of forecast accuracy. Methodologies are basically the vehicles for determining the consequences or implications of core assumptions that have been chosen more or less independent of the specific methodologies" [ASC78, p.199].

5.2 DIFFERENCES DERIVING FROM METHODOLOGICAL PURPOSE

In performing this technology assessment, it became increasingly evident that there are three[31] distinct "classes" of security-related evaluation methodologies. These classes are risk assessment, EDP audit and security evaluation. It must be emphasized that the boundaries between these classes are vague and overlapping. The classes are basically derived from the work performed by different professional communities in addressing security concerns.

As noted in Section 5.1, the first finding was that the methodologies in all three classes have striking similarities. Continued analysis, however, brought out increasing numbers of differences which seem to primarily derive from the perspective of the professional community involved. That is, EDP audits, security evaluations, and risk assessments, though all are sometimes viewed as forms of security evaluation, differ because of their purposes. It is important to note these differences. While they do not preclude a methodology developed for one purpose from being used for another, they can require substantial changes in the detailed implementation of the methodology. Lately, for example, there seems to have been an increased tendency to use risk assessment techniques for broader security evaluation purposes[32]. While the underlying security analysis structure readily permits (and even encourages) this, potential impacts as discussed below should be kept in mind. Table 5-1 provides a summary.

5.2.1 Basic Purposes And Elements

The basic purpose of traditional risk assessment is budgetary, with its objective being to optimally allocate security resources with respect to risks. Its emphasis has usually been quantitative, with risks generally viewed in terms of losses due to threats or attacks against system assets. Estimation of threat/attack frequency is a major step. According to FIPS PUB 65, "the essential elements of risk analysis are an assessment of the damage which can be caused by an unfavorable event and an estimate of how often such an event may happen in a period of time" [FIP65, p. 5]. The emphasis tends to be on who (threat) and what (asset) rather than on how (control).

---

[31] See footnote 29.

[32] FIPS PUB 65 [FIP65] is more oriented towards threats internal to a computer system than the older FIPS PUB 31 [FIP31] which is more installation oriented. Within DoD, NAVDAC has made substantial use of risk assessments for security certification.

Table 5-1. Differences Deriving from Purpose[1]

Risk Assessment	EDP Audit	Security Evaluation
1. Budgetary, optimally allocate resources	1. Assess controls, verify policy compliance	1. Assess defenses
2. Emphasize threats, assets, attack frequencies	2. Emphasize controls	2. Emphasize controls
3. Control existence and general effectiveness	3. Proper control functioning against anticipated threats/attacks ("within system rules")	3. Unanticipated ways to subvert or bypass controls
4. Balanced emphasis on exposures	4. Often emphasize modification (ensuring systems "tell the truth")	4. Usually emphasize disclosure (ensuring systems "protect secrets")
5. Controls considered last	5. Controls considered early	5. Controls considered early
6. Installation-oriented, often inadequate precision for systems and applications	6. Primarily application but also system oriented	6. All inclusive
7. Usually quantitative	7. Qualitative	7. Qualitative
8. Mutually exclusive exposures	8. Overlapping exposures	8. Often partial exposures
9. Balanced evaluation	9. Focus on key areas	9. Focus on key areas

[1] This table stresses differences in emphasis between traditional forms of the methodologies. It is not intended to summarize the complete scope of each "class" or address "hybrid" methodologies.

The basic purposes of EDP audits include checking for compliance with company policy and reviewing internal control systems [EDP80, p. 6.29]. Emphasis is also placed on assessing the quality of data (i.e. "substantive" auditing). Optimally allocating resources and evaluating the adequacy of security are usually seen as subsets of the auditor's role. Emphasis has been qualitative, with attention placed on controls, threats, and exposures. Threat/attack frequencies and assets are rarely considered explicitly.

Security evaluations assess defenses against hypothesized threats/attacks. Emphasis is on controls. Data quality is not a concern. Threats are also emphasized but tend to be detailed attack scenarios rather than generic events (such as fires). Threat/attack frequencies are not usually explicitly considered. Assets are of more concern than in auditing but of less concern than in risk assessment (e.g. they are not usually valued).

5.2.2 Analytic Emphasis

There are distinct differences in emphasis in the way the different evaluation classes view controls. Risk assessments are less explicit in evaluating the quality of individual controls and tend to be more concerned with their simple existence. Although the effectiveness of controls is considered in estimating losses, for the most part their proper operation is assumed. Little emphasis is placed on actually evaluating controls. This is especially true when assessing maliciously-generated risks associated with the functioning of applications and systems (as opposed to installations), since frequencies for such intentional attacks are difficult to determine. Even if these attack frequencies could be determined for applications and systems, it would be extremely difficult to determine how these frequencies would be affected by changing the internal control posture. Unlike fires and storms (e.g., natural installation threats), penetrators (e.g., malicious application and system threats) can go around strengthened controls. Alternatively, confronted with a greater technical challenge, they may even redouble their efforts to gain the greater satisfaction of penetrating increased controls. Therefore, interactions between interrelated controls and penetrator strategems must be analyzed before revising attack frequency estimates. Risk assessment methodologies do not readily support this analysis.

A widely held view is that most security violations, at least in the private sector, derive from accidental (not malicious) threats[33]. Here risk assessment is of greater applicability. In terms of defenses against accidental threats, the structure afforded by risk assessments may support general evaluations to determine the susceptibility of different components to flaws. Especially for applications and systems, however (where level of detail is greater),

they seem to be of quite limited value in identifying actual flaws (beyond the simple absence of required security functions). That is, for applications and systems, the lack of reliable threat and attack frequency data and the increased level of detail can result in risk assessments being of insufficient precision to serve as detailed evaluation methodologies.

Audits, like risk assessments, are concerned about control existence. Audits, however, place much more emphasis on the proper functioning of controls, in order to verify that the controls meet their objectives. Control objectives are often defined for this purpose[34]. The audit emphasis on proper functioning of controls has a key difference from the security perspective. Audits are concerned that controls function in accordance with the intent of the specifications and instructions. Lipner refers to this as "within the rules" of a system [LIP74] which is interpreted as meaning "within the context of threats and attacks anticipated by the developers and therefore countered by the system". In contrast, security evaluations are concerned that controls defend against threats in which the system is used in ways not anticipated or intended by its developers. Such "unanticipated" threats would not result in violations being recorded on an audit trail. Analysis of the two areas requires significantly different skills. The distinction between anticipated and unanticipated threats and attacks seems to be a primary distinction between EDP audits and security evaluations.

There can be some ambiguity here since auditors often view security as a subset of their responsibility while security people often view auditing as a subset of their responsibility. Much of this ambiguity is semantic. Both groups tend to take a narrow interpretation of the other's scope. When auditors speak of security, their primary exposure concern is generally disclosure (i.e. loss of sensitive data) and their primary threat concerns are often anticipated threats. Security people view auditing as "keeping records" in the sense of an audit trail.

---

[33] This view was expressed in the keynote address by Robert H. Courtney at the Working Conference of the Information Systems Security Association, Inc., March 28-29, 1985. It is quoted in their Quarterly Newsletter, Vol. 1, No. 2, June 1985.

[34] Much attention has been placed on control objectives in the audit and accounting communities [e.g. CIC70, EAF80]. An intriguing example is the Arthur Andersen & Co. evaluation approach which includes a lengthy treatment of control objectives [AAC78]. Compliance with the objectives is examined in a "general risk analysis". This use of the term "risk analysis" is intriguing since traditional risk assessments place little or no emphasis on control objectives.

In reality, the scope of both communities is increasingly overlapping. Security terminology and approaches grew from military and industrial beginnings with their roots in physical and personnel security issues. Especially with the introduction of computers, security concerns have grown to include the issues of data modification and service denial. Similarly audit terminology and approaches, from major roots in the financial community and its concern with data modification, have grown to include practically all aspects of verifying compliance with policy. (Note the significant difference between financial modification controls and disclosure controls, best illustrated by the fact that one cannot retotal sensitive textual material to detect disclosure violations).

This summary of audit and security emphasis has three points to make. First, there are terminology differences which can obscure the scope of each activity. Second, the scopes are increasingly growing and overlapping, although the audit scope remains much broader because of audit's interest in efficiency and effectiveness. Finally, the differences in primary emphasis remain. That is, in terms of internal controls, audits are mainly concerned with anticipated threats, and their major exposure emphasis is probably still modification violations. Security evaluations, while also concerned with anticipated threats, are very concerned with unanticipated threats, and their major exposure emphasis is usually disclosure violations.

Some of these differences among the three methodology categories can be illustrated by the way a password capability might be viewed in the different evaluations:

- Risk assessment - "The existence of an effective password capability results in reducing the attack frequency and the risk."
- Audit - "The password controls for authorizing the transactions were verified by the attempted use of invalid passwords."
- Security - "An application program was written which masqueraded as the executive and attempted to intercept the logon passwords."

The evaluation types also differ substantially in the timing of their control analysis. Risk assessments often do not explicitly consider controls (i.e. countermeasures) until an overall loss estimate has been obtained, at which point controls are factored into the process. Audit and security approaches explicitly consider controls much sooner.

Because of their differing purposes and characteristics, these evaluation types have been used in differing environments. Risk assessments, in the past, have been used primarily for installations; EDP audits for systems, with an emphasis on applications; and security evaluations for overall systems, with varying emphasis.

5.2.3 Quantification

Risk assessments are by far the most quantitatively oriented of the three types of methodologies (although recent indications are that risk assessments may be moving somewhat away from quantification). This stems from their concern with cost. While some audit approaches have quantitative aspects and may even (e.g. Peat Marwick Mitchell & Co.) result in a numerical rating, audit ratings are not absolute. Audit ratings tend to be relevant only within an organization or evaluation. The Annual Loss Estimates (ALE's) usually derived from risk assessments are in terms of dollars which theoretically should permit ready comparison across organizations. Security evaluation makes very little use of quantification. As a result of the stress on quantification, risk assessments generally quantify threat frequencies, asset values, and other factors which are usually dealt with qualitatively in audits and security evaluations.

In order to avoid counting costs twice, risk assessments also partition the evaluation process into mutually exclusive subsets. The best examples of this are exposure (i.e. impact) classes. In risk assessments, exposure classes are, to the first order, mutually exclusive (e.g. disclosure, destruction, data modification, denial of service) whereas in audits, exposure classes are loosely structured around overlapping management concerns (e.g. competitive disadvantage, statutory sanctions, loss of privacy data).

Also deriving out of this use of quantification in risk assessment is a more general and balanced evaluation of the entity in question. By comparison, audit and especially security methodologies focus their emphasis on key controls or areas of high potential vulnerability.

5.3 DIFFERENCES DERIVING FROM EVALUATION OBJECTIVE

The previous section discussed evaluation differences resulting from differences in purpose (i.e. risk assessment, EDP audit, security evaluation) of the underlying methodology. This section analyzes the impact on the evaluation process of differing evaluation objectives. Objectives discussed include the following:

- o Determination of actual flaws versus the susceptibility to flaws.
- o Detection of flaws associated with anticipated versus unanticipated threats/attacks.
- o Determination of proper operation versus acceptable development.

5.3.1 Flaws Versus Flaw Susceptibility

The objective of some evaluations is to detect system flaws (i.e. vulnerabilities) while other evaluations are concerned only with determining a system's susceptibility to flaws, and not finding the flaws themselves. Both approaches have benefits. Detection of flaws provides firm evidence for evaluation and certification. It permits correction of the flaws and improvement of security. It can also be used to confirm susceptibility findings and provide supporting examples. Determination of flaw susceptibility is extremely useful in helping to focus an analysis for specific flaws. Many methodologies incorporate a "preliminary" evaluation for this very purpose. Susceptibility determination also provides an intuitive feeling of confidence in the security defenses[35]. While this approach may seem to present overtones of an entity being "guilty until proven innocent", it can be a valuable approach depending on the purpose to be served.

Primary criteria used in establishing flaw susceptibility include developer objectives and methodology. For example, if the developers did not intend to provide security or if the development methodology was careless, flaw susceptibility would be high. The quality of other "certification evidence" [WEI78] such as system documentation also contributes to susceptibility determination.

In general, all of the methodologies support evaluations to determine flaw susceptibility. This is not the case with the detection of specific flaws. Here the methodologies differ with a primary distinction being the type of flaw involved (i.e. flaws associated with anticipated versus unanticipated threats and attacks). This is discussed below.

---

[35] The informal policy of one DoD agency is that the detection of specific flaws determines whether an entity can be certified whereas the more general issue of confidence determines what level of sensitive data should be entrusted to the evaluated entity.

5.3.2 Anticipated Versus Unanticipated Threats And Attacks

As described in Section 5.2.2, anticipated threats and attacks are those within the threat/attack context anticipated by developers and thus countered (whether successfully or not) by the system. Unanticipated threats/attacks are those in which the system is violated in ways not anticipated by its developers. The methodologies differ in their ability to detect flaws associated with these two classes. As noted in Section 5.2.2, risk assessment methodologies are useful in assessing the impact of the presence or absence of controls, especially installation controls, while not being highly applicable to detailed internal controls for systems or applications. Therefore risk assessments would be of some value in detecting flaws associated with anticipated threats/attacks since such flaws would primarily involve the general existence and effectiveness of controls. This value might be fairly limited, however. Traditional risk assessments would not be of value in detecting specific flaws associated with unanticipated threats/attacks.

Audit methodologies were typically designed for the evaluation of controls and the detection of flaws associated with anticipated threats/attacks, and are thus useful in that area. Their use of matrices seems to provide focusing via susceptibility determination, with checklists providing the critical audit and security insight, and (perhaps) transaction flow analysis the analytic approach to detect these flaws. Audit methodologies may be of limited use in structuring and supporting the search for flaws associated with unanticipated threats/attacks, but their supporting checklists would typically require significant extension to permit use for this purpose.

Security methodologies are sometimes designed for detection of both types of flaws (e.g. [AFI79], [NEM78], and the DoD and formal verification approaches). The SRI/ISI [NEM78] and verification approaches almost exclusively emphasize the unanticipated situations. Detection of these flaws is basically the process which has become known as "penetration". Matrices, checklists, and other methodological tools are of very limited use here. While checklists can provide security insight (representing "instant" experience and training), flaws associated with unanticipated threats and attacks are intricately woven into the specifics of the entity being evaluated. As a result, the best approach for detection of such flaws tends to be a somewhat unstructured immersion in design, interface, and procedural documentation.

This should not be taken to mean that there is no structure to "penetration" analysis, only that it is much less rigid than that normally associated with evaluations to determine flaw susceptibility or flaws associated with anticipated threats and attacks. There are in fact fairly accepted approaches to structure penetration analysis. For example, an analyst may look for flaws which fall into certain

flaw categories or patterns [HOL74, NEM78, WEB76] or may hypothesize generic flaws and then determine if they exist [LID75, WEI73].

Given the limited resources typically available for penetration analysis, it is usually most efficient and effective to concentrate analysis on the security boundaries, since these are the points at which attacks will occur. For applications, the boundaries tend to be represented as user interface procedures. For systems, where the primary security boundary is often internal to the system, the emphasis may have to be placed on design documentation. In both cases the use of critical walkthroughs is an excellent way to focus penetration attention.

In sum, risk assessment methodologies support general determination of flaw susceptibility and can be used to detect flaws associated with anticipated threats/attacks, especially for installations. Audit methodologies support detailed flaw susceptibility determination and fairly detailed detection of flaws associated with anticipated threats/attacks, especially for applications. Security methodologies support the detection of both types of flaws. Analysis for flaws associated with unanticipated threats/attacks is often referred to as penetration analysis.

5.3.3 Operation Versus Development

Security evaluations can be performed both during development and operation of the entity being evaluated. In-depth evaluations are much more appropriate for systems under development than for systems which have been in operation. There are several reasons for this. First, it is much more practical and less expensive to make changes (resulting from an evaluation) during development than during operation, and these changes usually have less negative impact on the organization. Also, funding for evaluation and changes would tend to be much more available during development. Of course evaluations are required during operation to assure that defenses are being maintained[36]. These would usually be more "test" oriented than evaluations to determine proper controls.

Historically, audit methodologies have usually emphasized operational evaluations. This is changing, however. A subcommittee of the President's Council on Integrity and Efficiency is jointly working with NBS on an EDP audit guide for FY'86 that treats the EDP audit role in system development. DoD security evaluation methodologies have historically emphasized developmental evaluation. The reason for increased emphasis in all communities on developmental evaluation is the growing realization that security controls (whether preventive, detective, or corrective) cannot be retrofitted into an inherently untrustworthy entity.

5.4 COMMON APPROACHES FOR STRUCTURING ANALYSIS

Economic forecasting is a form of predictive analysis which is critically relevant to our national health. It is also a field prone to substantial disagreement on analytical approach:

> "Tension exists between the approaches that attempt to capture the complexity of the economic system, and those that search for simple and immutable economic relationships or rhythms that would permit forecasters to cut through the complexity. Tension also exists between the approaches directed at the systematic improvement of methodology through the 'scientific method' of testing the result of explicit procedures, and the approaches employing judgment to capitalize on experts' experience and intuition" [ASC78, p. 59].

Perhaps it is to be expected that similar differences of opinion also exist in the area of security evaluation. Complex checklists and matrices are proposed against simpler transaction flow analysis techniques. High-level, general methodologies emphasizing expert judgment are proposed against extremely detailed ones (e.g. in risk assessment the high-level FIPS PUB 65 [FIP65] and Department of Agriculture [DOA77] approaches versus the detailed USAF AFRAMP [AFRAMP] and the SDC RAM [SDC79]).

The conclusion of this technology assessment is that the different approaches are not in general competing with each other. All have uses, advantages, and disadvantages with different ones being preferable for different people and situations[37]. Also it is often desirable to use several approaches in parallel or to combine approaches. The most commonly used approaches for structuring analysis are matrix, checklist, transaction flow, and loosely

---

[36] Threat of legal sanctions might reinforce this. As a point of conjecture, it would seem legally to be a more demonstrable (though not necessarily more severe) crime to inadequately operate a control than to fail to specify its existence. Similarly, failure to specify enough controls would be a more defensable position (e.g. there are no industry guidelines; the user, in knowing the controls, implicitly accepts the risk). This is an intriguing perspective, since it tends to view the presence of a control as a vulnerability, arguing for fewer, more reliable controls. Another conjectural legal perspective is on the evaluation itself. Legally, it would seem critical to document the evaluation process in detail since this could provide evidence of faulty analysis (however desirable this might be to other interests, including society's). These thoughts suggest counter-intuitive approaches which should be considered in a certification program.

structured (with loosely structured representing ad hoc approaches requiring highly experienced evaluators). Their advantages and disadvantages are summarized in Table 5-2 and discussed below. Other approaches exist which are not discussed here, such as structured (e.g. physical or functional) decomposition [HOF80, pp. 1, 2].

5.4.1 Matrix

Examples of the matrix approach include the Touche Ross & Co. methodology [MAI76] and Jerry FitzGerald's matrices [FIT78]. Matrices have a number of advantages as a tool to assist the security evaluation process. Their primary advantage is that they facilitate detailed structuring and partitioning of both the system being evaluated and the evaluation process itself. This helps in both understanding the process and in dividing responsibility for the work.

The use of matrices can also help to examine interrelationships. Touche Ross in fact uses one of its matrices solely for this purpose, with no physical product resulting, only improved understanding and perspective (see Table 4-3). Jerry FitzGerald's matrices, having already been completed at the generic level, provide substantial guidance into the interrelationships between controls, exposures, and assets. In a sense, his matrices, being completed, might even be viewed as a highly structured checklist.

Matrices help to ensure general structural completeness and as a result support high-level across-the-board evaluations. This results in their being quite useful for high-level flaw-susceptibility analysis but of limited use for the identification of specific flaws. They also are excellent records for documenting the evaluation process.

There are also disadvantages of the matrix approach. Some of those mentioned here were noted in Session 7 of the NBS workshop on Audit and Evaluation of Computer Security II [RUT80, pp. 9-11, 9-23].

---

[37] William Perry has identified six criteria to be used in selecting practices for auditing [PER78, pp. 1, 2]:

1. Computer audit practices must satisfy an audit objective or need in a given situation.
2. The auditor should possess the necessary skill level.
3. Auditors must have the resources to perform the practice.
4. The practice must be operational when it is needed.
5. The practice must be capable of being performed within budgetary limits.
6. The computer audit practice should be cost-effective.

Table 5-2 Comparison of Common Approaches for Structuring Analysis

Matrix	Checklist	Transaction Flow	Loosely Structured
Advantages			
1. Structure system and process	1. Ensure greater completeness	1. Focus attention	1. Fast
2. Help examine inter-relationships	2. Capture complexity	2. Improve perspective and understanding, address reason for controls	2. Simple
3. Ensure structural completeness	3. Heighten security awareness, perform training		
4. Document results	4. Document results		
Disadvantages			
1. General	1. Cumbersome and time-consuming	1. Doesn't structure overall evaluation	1. Litte documen-tation
2. No assumptions to support ratings	2. Lack of completeness		2. Require highly Exper-ienced personnel
3. No redundancy or sharing of controls			
4. Cumbersome and time-consuming			
5. Difficult to construct			

1. Because of their typical breadth, matrices tend to be very general. Additional detail is possible, but as FitzGerald has shown, the step to a lower level of detail would become very cumbersome with matrices because of the sheer numbers of components, controls, and other factors which influence the evaluation.
2. Usually matrices are structured so as to record analytical findings in the form of quantitative ratings or short phrases. This can be a disadvantage because the relationship between two entities is often not simple enough to be sufficiently represented by a rating or even a short phrase. Often it would be desirable to include, along with the rating, a discussion of the conditions or assumptions under which the rating applies. Matrices do not typically support such supplementary discussion.
3. Matrices also do not address redundancy or sharing of controls. This is certainly theoretically possible to do for very small portions of the system (e.g. via a combined matrix-decision table approach) but is probably not practical on any larger scale.
4. A major disadvantage of matrices for security evaluation is that they are extremely time-consuming to complete and can, by their cumbersome, rote nature, encourage shallow analysis. While some people enjoy lengthy, detailed, highly-structured, tedious work such as filling out matrices, others would undoubtedly find the process very difficult. Since both systems and potential security flaws can be highly complex, even seemingly simple matrices can require substantial time to complete. This is listed as a disadvantage of matrices but may in fact be more properly viewed as a characteristic of the security evaluation process in general which the matrix approach reveals, by its structuring.
5. Matrices, despite their apparent structural simplicity, are not simple to construct in an unambiguous way. For example, the categories in Jerry FitzGerald's matrices are concerns/exposures, resources/assets, and controls/safeguards. The use of slashes is illustrative of the difficulty in rigidly defining categories. The Touche Ross approach similarly notes the importance and difficulty of distinguishing between functions and controls. The point is that it can be very difficult to cleanly fit the elements of the security picture into their proper places. Unless this step is done well, the relationship between the elements in the matrices will be ambiguous and the resultant products will be more difficult to produce and of less value.

5.4.2 Checklist

The major advantages of checklists are in ensuring a higher degree of completeness, capturing complexity, heightening security

awareness, and documenting the evaluation process. While it is impossible for any checklist to be literally complete in listing all controls (for example), conscientious use of a good checklist will undoubtedly improve on a control study. There are many existing checklists of security evaluation elements such as controls, control objectives, threats, assets, and typical flaws. It would be useful to examine and incorporate many of these in tailoring a methodology for a specific organization. It would also be useful to have checklists of other factors, issues, and rules of thumb to provide guidance in implementing a methodology. For example, checklists would be useful in areas such as the following:

- What are the security policy alternatives and issues [e.g. JAC80, pp. E-5 through E-7]?
- What characteristics of assets or threats are relevant?
- What factors influence threat frequency?

These would be smaller lists than those mentioned above, but would be no less important. An analogy to illustrate the potential value of a small checklist is the familiar "who, what, why, when, and where" used for expository writing.

Most organizations with substantive security programs use checklists to aid their evaluations. The spontaneous, largely independent adoption of so many checklists is very significant. It suggests they might be a tool well adapted to both the evaluation task and human nature. There are two key advantages to checklists which reinforce this suggestion, namely, their benefits in (1)ensuring completeness and (2)capturing complexity.

The security "chain" has an essentially indeterminate number of links (i.e. controls). The more links examined during an evaluation, the better assurance there is that the chain is acceptably strong. Since it is impossible to examine every link, some approach is required to simplify the process. Checklists reduce the indeterminate length chain to a finite number of categories with varying numbers of representative elements (i.e. links) within each category. This permits the evaluation to attain the optimal amount of "relative" completeness.

As with matrices, a disadvantage of checklists is that they can be extremely time-consuming to use. Again, this is because they do not attempt to conceal much of the true complexity inherent in the security evaluation process. However, better than matrices, checklists can capture many of the subtle complexities and interrelationships inherent in security. This is because of their narrative nature. While matrices are somewhat bound to a flat structure and the use of single words or short phrases, checklists can use full sentences in all their complexity. This is not a trivial point. The ability to use complex narrative frees checklists from

being bound by their structure.

As a minor point, many checklists are oriented around questions [AMR71, AFI79, MAR73] while others are stated as requirements [HHS78]. The form is basically irrelevant although the use of questions would generally permit a lower level of technical detail.

5.4.3 Transaction Flow

As opposed to matrix-based analysis which is summary and structured in nature, transaction flow analysis is oriented towards the focusing of attention on a subset of the system. It is exemplified by the Arthur Andersen & Co. approach to financial auditing [AAC78]. The basic premise is that this focusing of attention results in increased perspective and understanding. The detailed products of transaction flow analyses might be seen as scenarios or examples. As a result, this can naturally complement the matrix approach just as detailed examples complement any general report.

Transaction flow analysis emphasizes the notion that security encompasses the dynamic flow of data (or any asset) through an interrelated series of controls and functions. It emphasizes that detailed understanding of the function being performed is absolutely critical to an assessment of security. That is, it is necessary to know the objectives and reasons behind the control in order to assess its adequacy. The intuitive appeal of a flow-based approach to understanding is reinforced by the SAC report, which uses a transaction flow organization as a way to structure an extensive list of applications systems controls [IIA77-1, IIA77-2]. This might even be seen as a sort of transaction flow checklist.

In situations where the primary concern is a small set of transactions or events, transaction flow analysis may almost suffice as a methodology. It would clearly be more applicable in audits and security evaluations than in risk assessments, which require a much greater element of completeness. Section 4.1.6.2 above also notes that transaction flow has been recommended as a reasonable way to evaluate application systems and the "desired" way to perform a data communication audit.

The primary disadvantages of transaction flow analysis are its potential lack of structure and completeness.

5.4.4 Loosely Structured

The DoD approach to formulation of an Evaluated Products List (see Section 4.1.4) essentially is based upon a loosely structured security evaluation by a team of highly experienced technical experts. Primary advantages of this are that it can be done very quickly and may be as accurate as any other approach (if not more so). Disadvantages are the lack of any documented process and the need for and expense of highly experienced personnel.

Theoretically it is possible (though unlikely) for individuals with little experience to use highly detailed methodologies such as USAF AFRAMP (for risk assessment) and achieve the same approximate accuracy as an expert using a loosely structured approach. There is little empirical evidence against which to determine whether this is so. Similarly there is no evidence to compare the accuracy of expert opinions using different approaches. It is probable that a systematic approach would be better than none and that a team evaluation (even if loosely structured) would be better than an individual evaluation. Even if evidence were accumulated to test these suspicions, the varying application-specific characteristics of each evaluation might make it difficult to make inferences about the applicability of the results to specific cases.

5.4.5 Focusing And Other Issues

William Perry quotes an EDP audit expert as stating that "auditing is 90 percent inefficient". This, Perry states, means that "auditors spend far too much time looking at the unimportant and too little time looking at what really needs attention [PER78, p. 2]". Clearly an extremely key part of a security evaluation methodology is its mechanism for focusing attention. Touche Ross uses a preliminary pass through its matrices to identify areas for more detailed analysis. Peat Marwick Mitchell uses a quantitative weighting done by a user group using the Delphi technique. The AFIPS checklist uses an organizational structure based around low, medium, and high risk issues. Loosely structured approaches use the expert judgment of the evaluators. There are also approaches independent of any specific evaluation methodology [HUB79, pp. 1-6], and also high level quick and general evaluation approaches (e.g., [IBM80]).

The USC/ISI work on protection errors takes this process one step further. Once an organizational component or control group has been selected for scrutiny, the evaluation process must continue to focus on more detailed categories. Indeed the USC/ISI work categorizing protection errors is based on experience which shows that "searches for errors are conducted most effectively by focusing on distinct well-defined types, one at a time, rather than by attempting to find errors of many different types all at once [CAR78, p. 1]."

The point of this discussion is that the focusing mechanism, while critically important, is independent of the methodology and can be adequately present in any generic structural analysis approach.

A few more additional issues are worth noting. One is that generic analysis approaches and detailed methodologies are often only vehicles for determining the consequences of underlying assumptions and are no better than the quality of the assumptions upon which they are based. For example, how much of a threat is really posed by wiretapping or operating system penetration? Will this change with the widespread introduction and use of small computers? How much degradation of service is acceptable? What would be the true impact of legal sanctions? The need to better define assumptions in such areas calls for a better balance between the development of more sophisticated evaluation techniques and the search for ways to establish and test the validity of such assumptions. Of course the less defined core assumptions are, the more experienced, team-based, loosely structured approaches would be preferable to highly structured ones.

The security evaluation field is a competitive one with emphasis continually being placed on greater specialization and complexity in methodological approaches. Demands of the discipline thus tend to force evaluation techniques to become both more technical and more ambitious. This increased analytical detail may exceed not only the assumptions as noted but also the utility to the user. In addition, there have been cases where increased attention focused on complex tools themselves has resulted in decreased attention to the actual evaluations.

5.5 GENERAL EVALUATION ISSUES

There are three important issues associated with security evaluation which are usually overlooked in discussions of methodology. These are quantification, uncertainty and bias, and integration with the decision process. All have the capability to invalidate evaluation findings if not properly considered. These issues are discussed at length below.

There are, of course, other issues which are also relevant. The human factors area is an example. It would include considerations such as performance variance among evaluators and the effects of boredom or motivation on performance. These are relevant considerations because they are influenced by the nature of the methodology. Other illustrative issues of concern in security evaluation are replicability and the ability to validate findings. Replicability refers to the fact that two teams using the same methodology to evaluate the same entity will often not obtain the same

findings. Ability to validate findings refers to the difficulties of determining the accuracy of an evaluation, even long after it has taken place. It also is related to the process of modifying the methodology, based on events, to improve subsequent evaluations. The point is that while quantification, uncertainty and bias, and integration with the decision process are important issues, many others also exist which need to be considered in formulating an evaluation methodology.

5.5.1 Quantification

Quantification is the determination or expression of an amount as is done in the process of measurement. There are two forms, absolute and relative. Absolute quantification includes numerical costs, frequencies, or other values which have meaning unto themselves. Examples are Annual Loss Estimates and threat/attack frequencies. Relative quantification includes numerical ratings, weights, rankings, and categorizations which require comparison or relation with other values and have no meaning unto themselves. Examples are the use of numeric levels for categorization (e.g. MITRE's "protection levels") and the generation of an overall security rating which has significance only within an organization or methodology (e.g. Peat Marwick Mitchell's "score"). Another example would be the theoretical use of "standard weights" [HOF77, p. 152].

Findings from this technology assessment indicate that many dangers are associated with quantification. In general, it seems the dangers are directly proportional to both the complexity of the element being "measured" and the extent of supporting data. The challenge with supporting data is to assess, accommodate, and reflect its quality. The accuracy and precision of the final product must reflect the accuracy and precision of the data upon which it is founded. The right question is preferable to the wrong answer.

There is fairly widespread use of relative quantification within evaluation processes. Typical purposes of this are to simplify the process and focus both attention and resources.

The methodology described in Auditing Computer Systems [PER80, pp. 5-16 through 5-19] uses one form of rating (e.g. Good = 5, Poor = 1) to simplify the evaluation of controls. The Touche Ross methodology [MAI76] uses a similar rating scheme but has found that even such limited relative quantification produces significant misunderstandings. Touche Ross is planning to remove the use of such numbers and replace them with letters to avoid tendencies towards misinterpretation of the results. The CICA methodology [CIC75] uses ratings (i.e., good, adequate) but requires that a narrative explanation accompany each. Fuzzy Risk Analysis uses linguistic terms

such as "high", "medium", and "low" for estimates (instead of numeric values). The explanation is as follows:

> "Numerical estimate values tend to fix the estimates at concrete values or ranges and remove subjectivity from the interpretation . . . . if subjectivity were present while making the estimate, it should be present during its review and evaluation" [HOF80, p. 3].

The conclusion here is that rating is a valuable aid in simplification but is susceptible to misinterpretation when numeric representations are used for the ratings.

The primary use of relative quantification within evaluation is to focus and balance attention and resources. Weighting is usually used for this although schemes such as ranking (e.g. prioritizing) are also used. While there is no empirical data which serves to evaluate the success of such focusing techniques, there seems to be a reasonable consensus that quantitative aids can be useful tools here.

Absolute quantification is seen more in risk assessments than in audit or security evaluations. Some forms of absolute quantification such as valuation of many types of assets or the estimation of frequencies of well-understood threats (e.g. natural disasters such as fire and flood) are generally accepted as reliable. This is because some assets, such as equipment, are easy to value and statistical data on many threats is available from the insurance industry.

Other forms of absolute quantification such as valuation of information (or the cost of reduced response time) or the estimation of frequencies of little-understood threats (e.g. wiretapping, subversion of a password scheme) are the subject of much discussion and disagreement. Glaseman, Turn, and Gaines conclude that this forces security evaluations to remain subjective [GLA77, p. 107]:

> "Risks to data and programs in a computer system are much more difficult to determine. There is very little experience in determining the value of exposed data files or programs, not all threats can be identified, and threat occurrences tend to be highly uncertain. Most importantly, empirical data about losses incurred by existing computer systems is virtually non-existent. Without data, attempts to evaluate system security must remain completely subjective."

They go on to state that what the risk assessments examined have done is to "try and describe what we would do if we really had all the knowledge and information about system vulnerabilities, people's intentions regarding those vulnerabilities, and exact dollar values concerning the losses that we could expect if particular attacks should occur"[38] [GLA77, p. 108]. Other communities, in confronting similar situations, have noted that the underlying data of greatest uncertainty can, as a result, assume the greatest importance [ASC78, p. 202]. The conclusion of Glaseman, Turn, and Gaines is that a much more detailed analysis is needed of the elements that contribute to security. In summarizing, they make the strong statement "we prefer to leave it as an open question whether or not a quantitative assessment methodology can ever be developed" [GLA77, p. 108].

This is reaffirmed by Campbell and Sands who state that a risk management model "should not require all factors to be reduced to quantitative terms" [CAM79, p. 294]:

> "[The] state-of-the-art is such that all factors cannot be reduced to discrete dollars and probabilities. Experience has shown that, except in highly specific situations, attempts to fully quantify all factors usually produce misleading results."

---

[38] Perhaps the risk assessment community could profit from the experience of the forecasting community which uses different forecasting techniques depending largely on the quality of the existing data which underlies the forecast. The following is from Sullivan and Claycombe [SUL77, pp. 33, 34]:

> "There are three basic types [of techniques] - qualitative techniques, time series analysis and projection, and causal models. The first uses qualitative data (expert opinion, for example) and information about special events and may or may not take the past into consideration. The second, on the other hand, focuses entirely on patterns and pattern changes, and thus relies entirely on historical data. The third uses highly refined and specific information about relationships between system elements, and is powerful enough to take special events formally into account. As with time series analysis and projection techniques, the past is important to casual models. These differences imply (quite correctly) that the same type of forecasting technique is not appropriate to forecast sales, say, at all stages of the life cycle of a product--for example, a technique that relies on historical data would not be useful in forecasting the future of a totally new product that has no history."

Much use has been made of quantification in other professions. Air pollution indices, economic indices, and even Scholastic Aptitude Test (SAT) scores have long been used in decision-making. In examining some of these usages, however, the limitations of quantification again become apparent[39],[40]. What also becomes apparent is that quantification is being used, in spite of its dangers.

There are advantages to quantification. According to the AFIPS Checklist, "some organizations have found scores, weight factors, and measures useful where the situation is well defined and well understood by all levels of management" [AFI79, p. 15]. The AFIPS document also notes that quantification can provide a basis for comparative analysis[41].

Probably the primary potential advantage of quantification is a characteristic which can also be its greatest disadvantage. That is, quantification greatly aids the promotion of the point of view which it represents. Ascher analyzes this powerful appeal of quantitative analysis and rating as illustrated by the popular attention given to the Limits to Growth model endorsed by the Club of Rome [ASC78, p. 35]:

> "Even if the argument was not original, it was for the first time 'demonstrated' by what appeared to be explicit, objective, scientific methods. The aura of science,

---

[39] A review of air pollution index systems showed that although "most respondents...expressed satisfaction with their own index....there was widespread opinion that the numbers expressed by indices are not necessarily meaningful" [OTT76, p. 8]. One agency discontinued its index "because the news media 'sensationalized' it by reading more into the index than was intended" [p. 9]. Despite this, however, and partly because of it, the Federal Government has sponsored the preparation of a standard index (the Pollutant Standards Index) [CEQ76].

[40] In qualifying the value of quantitative Scholastic Aptitude Test scores, the College Board "advises the colleges that the tests should not be overemphasized and that the test results should not constitute the sole basis for evaluating the probable future success of a candidate, but should be considered along with other relevant factors" [ANG71].

[41] It also adds that scoring and weighting may imply measurement of an organization. "In such a situation, managers may tend to withhold information or mislead the evaluation. This is especially true if failure to achieve a certain score will have a negative impact" [AFI79, p. 14].

> technology, and mathematics provided the plausibility that the core assumptions underlying the models lacked....modeling can enhance the promotion of the forecast by giving it the appearance of technical sophistication."

The conclusion of this technology assessment is that quantification is a useful but very volatile tool. It can be used to forcefully promote findings but can also serve to camouflage faulty assumptions or analysis. Management should be very wary of any security evaluation which results in only a final score. As it has been said, "good....judgment is superior to the 'numbers game'" [KRA79].

5.5.2 Uncertainty And Bias

Uncertainty and bias are subjects which have received almost no consideration in the literature associated with security evaluation methodologies. Since both can significantly affect any evaluation, much more attention must be given to this general area. Spetzler and Stael Von Holstein have written an excellent paper on this subject [SPE75]. Their analysis and findings, excerpted below, illustrate the relevance and importance of uncertainty and bias to the security evaluation process.

> "People seem to assess uncertainty in a manner similar to the way they assess distance. They use intuitive assessment procedures that are often based on cues of limited reliability and validity. Generally, these procedures or modes of judgment produce reasonable answers. For example, an automobile driver can generally estimate distance accurately enough to avoid accidents, and a business executive can generally evaluate uncertainties well enough to make his enterprise profitable. On the other hand, overreliance on certain modes of judgment may lead to answers that are systematically biased, sometimes with severe consequences....
>
> "Three features are worth noting: (1) Generally people are not aware of the cues on which their judgments are based. Few people know that they use haze to judge distances, although research shows that this applies to virtually everybody. (2) It is difficult to control the cues people use; the object seen through haze still appears more distant even when we know why. (3) People can be made aware of the bias, and then can make a conscious attempt to control its affects, as does a pilot when flying on a hazy day....

"For the purpose of this discussion the subject is assumed to have an underlying stable knowledge regarding the quantity under investigation. This knowledge may be changed by receiving new information. The task of the analyst is to elicit from the subject a probability distribution that describes his underlying knowledge. Conscious or subconscious discrepancies between the subject's responses and an accurate description of his underlying knowledge are termed biases....

"Biases may take many forms. One is a shift of the whole distribution upward or downward relative to the basic judgment; this is called displacement bias. A change in the shape of the distribution compared with the underlying judgment is called variability bias. Some discrepancies in distributions may be a mixture of both kinds of bias. Variability bias frequently takes the form of a central bias, which means that the distribution is tighter (has less spread) than is justified by the subject's actual state of information."

"Availability [probability assignments based on information that the subject recalls or visualizes] appears to be an important mode of judgment in most probability encoding sessions. It can also be introduced deliberately by the interviewer to help compensate for a subject's bias. For instance, if the interviewer believes that the subject has a central bias, he can ask the subject to make up scenarios for extreme outcomes, which thereby become more available and help counteract the central bias....

"There is a strong tendency to place more confidence in a single piece of information that is considered representative than in a larger body of more generalized information....People sometimes appear to assign probabilities to an event based on the ease with which they can fabricate a plausible scenario that would lead to the occurrence of the event....

"Subjects are seldom able to express their uncertainty in terms of a density function, a cumulative distribution, or moments of a distribution. Therefore, it is usually not meaningful to try eliciting a distribution or its moments directly. There are, for example, procedures that ask the subject for the parameters of a special distribution--for example, the mean and standard deviation of a normal distribution or a beta distribution. Our experience indicates that subjects will give such parameters, but that usually they do not understand the full implications....

"It should be clear that the encoding techniques discussed in this paper stress the interaction between interviewer and subject. We find that having the subject

assign a probability distribution without the help of an analyst often leads to poor assignments. This is true even for subjects who are well trained in probability or statistics. The main reason for our emphasis on interaction is that it is difficult to avoid serious biases without having an analyst present."

Several conclusions can be drawn from the excerpts:

1. Inaccuracies deriving from uncertainty and bias are of strong relevance to the security evaluation process.
2. There exists a body of research on uncertainty and bias.
3. Techniques exist to anticipate and offset bias (e.g. the authors go on to list uncertainty principles and provide procedural interview guidance).
4. Estimates based on uncertainties should include probability distributions[42].
5. Individuals can not reliably state the confidence of their own estimates.

To some extent, the evaluation community has taken steps to address uncertainty and bias. Some of the methodologies do provide guidance in interviewing [AFI79, MAI76, PER80] but this is typically very limited and shows no systematic recognition and treatment of the two areas. Most such guidance is provided in the audit methodologies. One exception is the apparently fairly extensive interview guidance that was to be provided with the Department of Agriculture methodology [DOA80].

One systematic way to accommodate uncertainty and bias, as well as the highly complex and poorly-understood nature of security evaluation in general, is to use the Delphi method. This is especially true if uncertainty about underlying assumptions negates the value of detailed methodologies. The traditional approach to decision making under such complex situations is to obtain expert opinion through open discussion and to attempt to arrive at a consensus among the experts. Unfortunately, results of panel discussions are sometimes unsatisfactory because group opinion is highly influenced by dominant individuals and/or because a majority

---

[42] This can be illustrated with another example from forecasting. According to Sullivan and Claycombe, "forecasts should be two numbers" [SUL77, p. 2]. "Since forecasts will more than likely be incorrect, it is vital to have some estimate (itself a forecast) of how wrong it can be. Any forecast that does not indicate a range, for example + 15% or 2300-2800, is only half a forecast. There is no excuse for lacking some measure of accuracy in the statement of a forecast."

Some of the methodologies examined here have mechanisms to reflect such "confidence" (e.g. [HOF80, p. 5], [SDC79]).

opinion may be used to create a "bandwagon" effect [SUL77, p. 140].

The Delphi method is a systematic technique for soliciting and organizing expert opinions through the use of anonymous, iterative responses to a series of questionnaires, and controlled feedback of group opinions. The two basic premises are that (1) persons who are highly knowledgeable in a field have the most plausible opinions and (2) the combined knowledge of several persons is at least as good as that of one person.

While the Delphi method has a number of pitfalls[43], it has potential as a useful technique which should be considered for wider applicability in the security community. Peat Marwick Mitchell uses the Delphi technique to weight its ten security analysis areas, in order to focus attention and produce a final quantitative score. The DoD community performed a pseudo "Delphi Forecast" on Trusted Operating Systems [AF79].

One conclusion that is evident from the above discussion is that the field of decision theory (of which studies of uncertainty, bias, and the Delphi method are subsets) is highly relevant to the security evaluation process[44]. Nielsen and Ruder support this, in concluding that "the theory of decision making under uncertainty is a fertile area for those working in the field of computer-system security" [NIE80, p. 25]. The relevance of decision theory is perhaps best shown by Spetzler and Stael Von Holstein's summary of the decision analysis framework [SPE75, pp. 341,342]:

---

[43] Linstone lists eight [LIN75, pp.571-586]:

1. Discounting the Future (i.e. not sufficiently accounting for future events.)
2. The Prediction Urge (i.e. suppression of uncertainty).
3. The Simplification Urge (e.g. "complex systems frequently exhibit strongly counter-intuitive behavior.")
4. Illusory Expertise (i.e. failure to recognize the applicability of major areas of expertise).
5. Sloppy Execution (e.g. lack of imagination, impatience).
6. Optimism - Pessimism Bias (e.g. long range overpessimism; short range overoptimism).
7. Overselling (e.g. improper use).
8. Deception ("The Delphi process is not immune to manipulation or propaganda use.")

[44] Another example to reaffirm this is work by Henderson and Nutt which explores the effect of decision style on the decision maker's perception of risk [HEN80]. Sage and White pursue the subject further, noting the influence of differing "actors" associated with decision situations involving risk [SAG80, p. 428].

- o Deterministic phase
  - Define relevant variables
  - Characterize their relationship in formal models
  - Assign values to possible outcomes
  - Measure importance of different variables through sensitivity analysis

- o Probabilistic phase
  - Explicitly incorporate uncertainty by assigning probability distributions to the important variables (distribution obtained by encoding the judgment of knowledgeable individuals).
  - Process judgments using models from above, resulting in probability distribution that expresses the uncertainty about the final outcome.
  - Factor in decision maker's attitude about risk, establish best alternative.

- o Information phase
  - Compare value of added information (to reduce uncertainty) with the cost of obtaining it.

5.5.3 Integration With The Decision Process

Related to the above issue is the third evaluation issue of integration with the decision process. A primary role of management is to make decisions. Decisions involve risk. Management is thus responsible for taking organizational risks. Security risks are only one form of the risks that must be taken. The key issue, then, is how other organizational decisions involving risk are made. Security risk decisions must take place within the same framework.

There are two aspects to this integration: evaluation content and presentation of findings. The methodologies examined tend to be quite weak in both areas. In terms of content, the evaluation must be keyed around information that management needs, understands, and most

importantly, uses. This will influence evaluation objectives, scope, resources, and emphasis. The careful selection of exposure categories as discussed in Section 3.3 is important here. In terms of presentation, it is critical how findings are coordinated within the organization and recommendations presented. Braithwaite has developed a procedure (oriented around a questionnaire) for presenting recommendations to management [BRA80].

5.6 CONCLUSIONS

Dorothy Denning summarizes the problem of securing statistical data bases as follows [DEN78, pp. 525 and 529]:

> "Recent studies reveal that the problem is even more difficult than at first believed. Methods once thought to significantly reduce the threat of compromise....are in fact easy to circumvent.... Other techniques...may be robust, but at the price of limiting the usefulness of the data base. The conclusion is that complete privacy cannot be enforced without severely restricting the free flow of information. The questions of interest then become:
>
> - Can we measure the relative security of a data base?
> - What is an acceptable level of security?"

This example serves to illustrate the basic dilemma confronted by security evaluation.

The audit community, confronted with similar problems, has met with a similar lack of success. William Perry, in assessing computer audit practices, writes that "the computer audit techniques that are used are, in reality, sophisticated ways of 'auditing around the computer'.... none of them involve an audit analysis of the adequacy of internal controls within a computerized application" [PER77, p. 9]. He reports data processing managers feeling that "computer auditors are too simplistic" and that "auditors' control requirements are too rigid" [PER76, p. 3].

In the fields of computer security and risk assessment, Glaseman, Turn, and Gaines stress that "progress in security assessment depends primarily on the use of more precise information" and conclude that "it is premature to search now for a global security assessment methodology" [GLA77, pp. 108 and 111]. They suggest two major requirements for progress in security assessment [GLA77, p. 111]. These requirements remain very valid:

1. "Increased research emphasis aimed at the development of a better understanding of the informational elements of security assessment."

2. "Experience, at the level of individual computer installations, in the application of a broader and more accurate information base to the assessment of computer security."

Based on all the investigations, readings, and discussions that accompanied the production of this technology assessment, the authors of this report have drawn their own set of conclusions about the state of the art of measuring the level of computer security. These are as follows:

5.6.1 On Measuring Levels Of Computer Security

The conclusion of this technology assessment is that there is no widely accepted existing way to measure a level of computer security. This derives from both the lack of generally acceptable "levels" and the lack of precise "measurement" capabilities. All top security experts consulted were pessimistic about near-term breakthroughs. Existing methodologies are useful, however, in guiding and structuring security evaluation and can produce meaningful results (or misleading results if improperly used).

5.6.2 Need For Guidance On Security Evaluation

There is a strong need for guidance in security evaluation. This conclusion stems not so much from shortcomings in existing methodologies as from the much more widespread lack of evaluation programs and capabilities in government and industry. In the process of formulating this guidance, there is much to be learned from the general approaches and experience of the audit and risk assessment communities.

5.6.3 On Selecting A Methodology

Existing evaluation methodologies all have advantages and disadvantages with different ones being preferable for different people and situations. In a single evaluation, it will often be desirable to use several different approaches in parallel for different organizational or system components. It will also be desirable to combine approaches to form hybrids. The underlying purpose of the evaluation methodology (i.e. risk assessment, audit, security evaluation) and the objectives of the specific evaluation (e.g. detect flaws versus flaw susceptibility, detect flaws associated with anticipated versus unanticipated threats/attacks) greatly influence the nature of the evaluation process.

5.6.4 On Selecting The Purpose Of The Methodology

The main purposes of a security evaluation methodology are to structure, guide, and record the analysis. A methodology should ensure awareness of the existence of and relationship among all relevant information elements. A proper balance must be struck between the judgment embodied in the methodology and the experience required to use it. This balance will differ for different organizations.

5.6.5 On Tailoring The Depth Of The Evaluation

Since any set of security defenses can be penetrated given the expenditure of enough resources, it would be possible to expend commensurate resources for security evaluations. This is clearly not feasible. The solution lies in tailoring the depth of the evaluation to the situation and resources (both monetary and personnel). In-depth evaluations are much more appropriate for systems under development than for systems which have been in operation. Of course some evaluations are required during operation to assure that defenses are being maintained. These would be more "test" oriented than evaluations to determine proper controls.

5.6.6 On The Importance Of Skilled Evaluators

In concluding this section, it is important to stress a cautionary note. The most critical need in performing a security evaluation is the use of people who have sufficient motivation, intelligence, security perspective, and knowledge of the entity being evaluated to perform the work. Methodologies can help provide training and perspective and can guide the work, but people must still do the work. No methodology, no matter how detailed, can supplant the need for judgment, common sense, and hard work. Indeed, the use of a detailed methodology can easily impede evaluation if it diverts attention from the basic analysis task at hand. For some evaluations, a sufficient "methodology" may essentially be to review user manuals with common penetration approaches in mind and run tests in likely problem areas. With any methodology the most important need is to adapt it to meet the management, organization, and situation-specific characteristics, resources, needs, and objectives. The method must not obscure the mission.

CHAPTER 6

SECURITY POLICY IMPACT

There are no absolute "units" of security. As a result, security "measurements" must be stated in relative terms. The background and scale against which this measurement takes place is that defined by security policy. A technology assessment on security evaluation would not be complete without an examination of policy impacts on the evaluation process. Security policy discussion is divided into two components: sensitivity distinctions and acceptance criteria.

6.1 SENSITIVITY DISTINCTIONS

Sensitivity refers to the consequences of loss. The greater the potential loss if data (or equipment) is disclosed, changed, destroyed, or delayed[45], the greater the sensitivity of that data (or equipment). It follows that the greater the sensitivity involved, the greater the need for security. Of course the need for, say, increased application security might be satisfied by system or installation controls, not application controls.

There are two generic structural types of sensitivity distinction: horizontal and vertical. Both are typically used together. Horizontal distinctions are sometimes referred to as communities of interest. Examples of such horizontal categories for data include proprietary, financial, personal, medical, psychological, political, investigatory, system controls, test answers, and national security (perhaps further broken down as plans, capabilities, and intelligence sources). Vertical distinctions are sometimes referred to as security or sensitivity levels. Examples are Top Secret / Secret / Confidential / Unclassified as used in DoD and the Office of

---

[45] Sensitivity is often interpreted as applying primarily (or only) to disclosure losses. This can be very misleading. Some organizations (e.g. many banks) regard data integrity (resistance to data modification or insertion) as the primary type of loss; others (e.g. the Social Security Administration) regard denial of service as the primary loss category.

Personnel Management's (OPM) ADP I (Critical-Sensitive)/ADP II (Noncritical-Sensitive)/ADP III (Nonsensitive) structure which might incorporate national security, privacy, life critical, automated decision-making, and other information types [EPP80, pp. J-13 through J-16].

As one might expect, categorization of data can be very difficult. This is well shown in the distinctions among the national security information classification levels where Top Secret, Secret, and Confidential data are distinguished, respectively, based on whether their disclosure causes "exceptionally severe damage", "serious damage", or simply "damage" to national security[46]. Issues such as data aggregation can also complicate matters. (This refers to situations in which a large group of, say, Confidential elements would be classified as Secret even though no single element was Secret.) In some cases, it may not even be feasible to establish a classification scheme. This was the case, for example, at General Motors (GM) [JAC80, pp. E-4, E-5].

> "In GM and most private sector companies, the words 'sensitive' and 'critical' are informally defined as classifications of data.... The formal classification of data requires identification of formal procedures for handling each class of data, a requirement for individual clearance for a given class of data, and the assignment of the classification to each data element. Little utility is seen in the private sector for this formal approach to classification. [A study was conducted within GM] to establish standard classes for both information systems and data. This was to be done so that we could say that a given application is of a given class and hence should be handled with the rules for that class....The study found no practical economic and effective classification scheme."

In some cases access control capabilities may be required at a finer level of granularity than that supported by an organization's formal sensitivity distinctions. This gives rise to what has been referred to as a "data dependent" protection policy [GUD80]. For

---

[46] Unusual relationships between the clearances which allow access to this data can also complicate matters. For example, Restricted Data is a horizontal category defined by the Department of Energy (DoE). Restricted Data is further vertically divided into the national security information levels. A DoD security clearance (e.g. Secret, Confidential) has been accepted by DoE as the basis for access to Restricted Data of the same or lower classification. There exists, however, a DoE "L" clearance which allows access to DoD data up to the Secret level but Restricted Data only up to the Confidential level [NAV79, p. 17-10].

example, a user may be restricted to see only records with SALARY <20,000 and another user restricted to see only records of department X. This type of policy is viewed as a special case of the two defined above (rather than as a separate type of sensitivity structure) and is for the most part controlled via discretionary access policies as discussed below.

Access policies associated with sensitivity distinctions are either mandatory or discretionary. The difference lies in whether the data holder (creator or owner) has discretionary authority with respect to who can share the information. Mandatory access policies (e.g. those relating to DoD's security levels) cannot be overruled by the data owner[47]. Discretionary policies (e.g. defining who can access which Secret or financial data and what capabilities are associated with the access, such as read, write, extend, delete) are defined by the data owner within the constraints of the mandatory policy. Lee et al note that discretionary policies are "what most systems have today" [RUT80, p. 8-12].

Essentially all organizations which have sensitivity distinctions use a horizontal structure with many also using a vertical structure. Since security is basically the process of defining and enforcing segregation, it should be theoretically possible for all organizations to suffice with only a horizontal structure. For example, since Top Secret and Secret data are at separate levels, they could be represented horizontally rather than in a vertical relationship. Where levels are used, however, they can serve to define levels of protection requirements in addition to levels of potential loss. For example, Confidential, Secret, and Top Secret clearances (for access to the corresponding data) require distinct and increasingly thorough background investigations.

DoD has several other interesting structures for levels of protection requirements. One is the operational security mode. DoD Directive 5200.28 defines four such modes (dedicated, system high, controlled, and multilevel) which impose increasingly stringent protection requirements on the ADP system in question [DOD78]. Another is the hierarchy of protection levels described in Figure 4-3. These structures are not inherently related to any vertical or horizontal sensitivity distinction scheme[48]. They are, of course, strongly relevant to security evaluations.

In general, there does not seem to be a strong correlation

---

[47] There are exceptions even here, of course. For example, Navy security policy provides specific exceptions to mandatory policy in some cases involving such persons as retired officers or former presidential appointees [NAV79, p. 17-19,17-20].

between security evaluation and the organizational approach taken for sensitivity distinction. The primary relationships which do exist are as follows:

- The greater the sensitivities involved, the greater the emphasis typically placed on security evaluation.
- Different organizations may be responsible for different types of sensitive data or equipment, with resultant different evaluation policies[49].
- Different types of sensitive data and equipment may have different protection requirements[49].

6.2 ACCEPTANCE CRITERIA

Acceptance criteria [NEU82] can be thought of as specialized security requirements. They are specialized because they represent a different perspective from other security requirements. That is, whereas normal requirements are typically formulated in response to the question "what do we need?", acceptance criteria respond to the question "how will we decide if the product is acceptable?" These are clearly overlapping sets since products are usually defined as being acceptable if they meet needs. The problem is that if only the first question is asked, needs will probably not be sufficiently defined. The role of acceptance criteria then is to ensure that the requirements include sufficient definition of:

- What degree of quality (e.g. performance, penetration resistance) is required?
- What will be examined in evaluating the degree of quality?

Acceptance criteria thus should be "measurable" or demonstrable

---

[48] The Air Force Summer Study work, however, defined relationships between protection levels and the degree of difference in the number of security levels being simultaneously supported within a system [AF79, p.85].

[49] Uniform sensitivity distinctions (e.g. in the federal community) would simplify the provision of centralized policy guidance and might permit the guidance to be more detailed. For example, uniform sensitivity distinctions might facilitate the categorization of government systems and the establishment of control objectives or requirements for the different categories. As with any form of centralized "regulation", this has both good and bad aspects.

features of required security functions which characterize their desired quality. They serve as decision criteria which are used to determine whether a product complies with security requirements.

6.2.1 Definition Of Acceptance Criteria

Definition of acceptance criteria is the most important task associated with security certification. This is so because it underlies and often defines both the design of a system and the associated certification process. Certification, after all, is based upon the process of determining whether a system satisfies a set of acceptance criteria (i.e. complies with security requirements).

Definition of acceptance criteria is also the most difficult task associated with security certification. This is because the process of defining acceptance criteria is:

1. Subjective
2. Complex
3. Dynamic
4. Based on little experience

Each of these is expanded upon.

1. Subjective. It is commonly stated and accepted that absolute security is not achievable. This is true. There are, for example, no absolute defenses against human subversion, human error, or hardware failure. The implications of this situation on acceptance criteria are major. It is not meaningful, for example, to say "the system must prevent data disclosure", when it is impossible, even with unlimited resources, to absolutely ensure this prevention. This situation has forced those defining acceptance criteria to examine alternative approaches such as defining levels of security or specifying acceptable rates of loss. Unfortunately, our research has detected little success in either area. The primary reasons behind this lack of success seem to be complexity of the entities and unavoidable probabilistic elements. The result is that the definition of acceptance criteria remains a very subjective process.

   Because of this and because no widely accepted acceptance criteria exist, the establishment of criteria tends to be a very judgmental, often apparently arbitrary process. A real and forceful illustration of this process is

the description below of how the National Aeronautics and Space Administration (NASA) defined the overall reliability acceptance criterion for the Apollo program [OTO79].

> "When President Kennedy gave birth to Apollo, some of the best minds in the country were giving it one chance in ten of making it to the moon. But [NASA] engineers were choosing much better odds: 999 to 1. Caldwell Johnson, an engineer at the Manned Spacecraft Center in Houston remembers how the odds were chosen.
>
> 'The question of reliability came up', Johnson said not long ago. 'Should 50 percent of the missions be successful? Should 9 out of 10 guys come back alive?
>
> 'Or should it be 999 out of 1,000 guys? The cost of development is a function of reliability. If you can afford to lose half the spacecraft and half the men, you can build them [much] cheaper.'
>
> While work on the Apollo design stopped in 1961, the question was debated for weeks. With nobody willing to make a decision, the engineering team turned to Robert Gilruth, then director of the Manned Spacecraft Center. Engineer Max Faget spoke up: 'If we're successful half the time, that will be worth it'.
>
> 'No, that's too low', Gilruth said. 'We can make 9 out of 10. Maybe 99 out of 100, lose one man out of 100 on lunar missions.'
>
> 'That's ridiculous', said Walt Williams, the director of the oneman Mercury. 'Make it one in a million.'
>
> 'How about three nines?' Gilruth responded. 'How about a reliability of 9-9-9?'
>
> And so it was."

This example illustrates well the difficulty in defining acceptability and also reveals the nature of the subjective, qualitative foundation upon which quantitative criteria are often based.

In addition to its subjective nature, there is another facet to defining security acceptance criteria which complicates the process. That is, not only is absolute security unattainable, but also threats exist for which there are no defenses. In a sense, these are two ways of saying the same thing. The trouble with the customary "absolute security" discussion is that it does not state the problem strongly enough. The situation is graphically described by Lipner [LIP74, p. 2].

> "The basic problem . . . is that any program that runs on a computer can access any information physically accessible to the processor, and can [without detection] retrieve, alter, or destroy the information as the programmer wishes . . . . [In addition], if an error in an operating system program allows a penetration program to work, that program will work every time it is executed . . . . The probability of a successful penetration is then unity; the level of security zero percent."

It is probably still true that no existing operating system can prevent a bright, malicious, highly-motivated user programming in assembly language from gaining control of the system. No audit scheme has been devised which could not be bypassed by such a user. No investigation process has been devised which could identify all malicious users (not to mention all bright, highly-motivated ones). The problem, then, is not just that absolute security cannot be achieved, the problem is that it is absolutely certain that any defense can (though not necessarily will) be penetrated. This is an important consideration in the definition of acceptance criteria.

2. Complex. While standards or regulations may impose requirements for certain security functions or features, ideal theoretical security requirements would be phrased in more generic terms, such as acceptable rates of loss or degradation. The reason is that requirements so stated would not arbitrarily constrain designs by imposing the need for certain mechanisms. Unfortunately, attempts to achieve this theoretical ideal are severely frustrated by the complexity of the security problem. Security involves the protection of many forms of assets against many types of loss from many different threats. The complexity of this situation is shown by Figure 6-1, which depicts an illustrative attempt to define acceptable rates of loss or degradation[50a,b].

	System Reliability				Disasters		Malicious Acts			
	Hard-ware	Soft-ware	Comm-unica-tion	Appli-ca-tions	Human Error	Natural	Financial	Users	Opera-tors	Devel-opers	External Personnel
Disclosure					1/10,000 Trans-actions		Not revealable by subpoenas	1/50 Att-empts			
Sensitive Data					1/100 Trans-actions			1/20 Att-empts			
Other Data						Fire, Flood, etc. $10K/Yr					
Destruction					$2K/Mo.						
Building											
Equipment	$1K/Yr				$2K/Yr						
Software											
Information											
Data Integrity			1 in 10^5 bits	1/1000 Trans.							
Modification (Accuracy)											
Insertion											
Denial of Service								10 sec.			
Response					3 sec.						
Throughput											

Figure 6-1.
Illustrative Definition of Acceptable Rates of Loss or Degradation

---

[50,a] Note the separate treatment of accidental and intentional threats, adapted from [AFI79]. Parker states that more widespread adoption of this type of approach is a major requirement for progress in security assessment [PAR78].

Several illustrative criteria have been included. In actuality, such a simple matrix would be grossly inadequate for these purposes. In the example, practically every entry on each axis requires subdivision (often extensive). Also, each criterion requires discussion in significant detail (as well as probability ranges for each quantity). That is, for each criterion, questions such as the following should be answered:

1. Under what conditions (cost, collusion, access, system state)?
2. With what likelihood of detection?
3. Would losses be total, partial, intermittent?
4. Is the criterion conditional on one in another area?

Because of these difficulties, rates of loss or degradation are not generally used as acceptance criteria. Instead, criteria such as control objectives, control functions and acceptance tests are used. While reducing complexity, these approaches still are subjective and require large amounts of judgment in evaluating compliance.

This issue of evaluating compliance is a key one and one which is aggravated by the inherent complexity which remains. Acceptance criteria are only as good as the capability to evaluate compliance with them. Indeed, the existence of a capability to evaluate compliance (e.g. performance measurement) must itself be an acceptance criterion. The most commonly used criteria to date have been those which can be most easily evaluated. Examples include the ability to perform specific functions (e.g. predefined acceptance tests) or the existence of specific control features (e.g. audit community control guidelines). Another form of criterion which has been used is the error rate, with an example being the bit error rate on communication lines. This can be evaluated through testing or, in some cases, analysis based on hardware/software failure rates and the number of components. As discussed at length in Section 4.1.5.2, theoretical work in measures of coverage and software quality metrics is also applicable in this area, as is formal verification. One problem with these forms of

---

[50,b] Disasters can be structured in many ways. The Defense Civil Preparedness Agency lists natural and man-made disasters. Natural disasters include forest fires, hurricanes, floods, tornadoes, winter storms, and earthquakes. Man-made disasters include plant fires, chemical accidents, transportation accidents, public demonstrations and civil disturbances, bomb threats, sabotage, radiological accidents, and nuclear attack [DCP78].

criteria is that, except possibly for formal verification, they provide little insight into resistance against malicious attacks.

3. Dynamic. The acceptability of criteria will change with changes in technology and management. With respect to technology, what is acceptable depends upon what capabilities are possible and available. Both are constantly changing and often misrepresented. With respect to management, it is the responsibility of new management to reassess and perhaps change the security program. Security requirements or acceptance criteria would be a likely area for changes reflecting differing perceptions, priorities, and styles. As above, it must be possible to evaluate compliance with any proposed new criteria. Further system changes might be needed to permit this evaluation.

4. Based on Little Experience. The task of defining acceptance criteria for computer security is a new one. Although the process, by its subjective, complex, dynamic nature, demands extensive experience, little such experience is available. Criteria should improve as more experience is gained in their definition.

6.2.2 Security Requirement Classes For Evaluation Analysis

A key to the entire subjects of both acceptance criteria and evaluation is the fact that there are several distinct classes of security requirements. This is key to the understanding of acceptance criteria because the different requirement classes define the different corresponding classes of acceptance criteria. It is key to the understanding of evaluation because different skills and techniques are required in evaluating for compliance with the different requirement classes. The requirement classes, along with corresponding evaluation classes, are shown in Figure 6-2. Each class is clarified by a question which illustrates its primary intention.

This structure for security evaluation classes accommodates the differing objectives discussed in Section 5.3, such as finding flaws versus susceptibility and finding attacks not anticipated by developers. (Of course within each class, evaluations would still have to be adapted to satisfy lower-level objectives, such as an emphasis on disclosure versus integrity or overlapping versus mutually-exclusive losses.) The applicability of different evaluation methodologies examined in Chapter 4 is discussed for each class, in order to integrate the analysis in this technology assessment.

Requirement Classes (What is needed?)	Security Evaluation Classes (How well do defenses meet the requirements?)
1. Functional (What functions are required?)	1a. Control Existence (Are the required functions provided?) 1b. Functionality (Do they work?)
2. Performance (How well should they work?)	2. Performance (How well do they work?)
3. Penetration Resistance (How strong should they be?)	3. Penetration Resistance (How easily can they be broken or bypassed?)
4. Methodological (How should they be built/ acquired?)	4. Methodological (How were they built/ acquired?)

Figure 6-2. Requirement/Evaluation Classes and Objectives

1. Functional. This class is concerned only with the provision of security functions. Many existing acceptance criteria and evaluation methodologies do not progress much beyond concerns in this class.

Methodologies examined in Chapter 4 can be very useful in defining functional requirements. Perhaps the most applicable techniques are risk assessment methodologies, since these seem to have been developed primarily for the task of requirements definition. As implemented in a security context, risk assessments are essentially cost-benefit analyses for security. Their best use is for installation-oriented requirements definition, although they are sometimes used to structure analysis of applications and systems requirements. Audit and security evaluation methodologies can also be useful in functional requirements definition. This is because the question of how much we need (as addressed in requirements definition) is very similar to the question of how much we have (as addressed in the evaluation). Existing evaluation methodologies can help to both structure and implement this analysis.

The purpose of an evaluation for control existence is to determine whether the required security functions have been implemented. This involves determining that requirements have been implemented in the specification and that specifications have been implemented in the product. In some situations requirements may not have been well defined, or specifications may not exist. Both of these situations will extend the scope of the evaluation activities needed to determine control existence. In other situations (e.g. where well-defined requirements and specifications exist), control existence determination will not require a full evaluation process. Simple, almost clerical correlation between the requirements and specification may be sufficient. Of course, on the completion of development, testing will often be required to verify findings.

The purpose of evaluations for functionality is to determine whether controls will work as intended. The major concerns are design or implementation errors. Both design review and especially testing are useful here. Traditional forms of audit testing and acceptance testing should typically suffice.

2. Performance. Performance issues include such factors as response time, throughput, accuracy, availability, and survivability. Evaluations for the performance of controls can be very difficult and require high degrees of experience

and judgment. To a certain extent, the requirements definition phase can assist by defining measurable requirements where possible and also requiring that measurement capabilities be built into the entity. In many cases, however, it will not be possible to define acceptance criteria measures for performance.

Evaluations of this sort require both design review and testing activities, with primary emphasis on testing. In general, except for testing, detailed evaluation methodologies are not of substantial use here.

3. Penetration Resistance. The major questions addressed here are whether the controls can be broken (e.g. as in the case of passwords or encryption algorithms) or bypassed (where penetrators go around a control). Resistance applies to attacks against the entity, data, or performance. As with evaluations for performance, evaluations for penetration resistance are extremely difficult and can require much experience and judgment. Centralized evaluation (as in the Department of Defense), security penetration (e.g. [HOL74, NEM78, WEB76, WEI73]), and formal verification are applicable approaches.

4. Methodological. The methodology used to develop or acquire controls is a key facet of control quality. From the requirements perspective, a higher quality process increases chances for a high quality product. From an evaluation perspective, development or acquisition methodologies are important in determining flaw susceptibility and general confidence in the product. In addition, flaws in the process can create flaws in the product. Existing evaluation methodologies do provide some guidance in evaluating the development process (e.g. [MAI76], [NEM78]).

6.2.3 Types Of Acceptance Criteria

At this point we have examined the difficulties of defining acceptance criteria and the different requirement classes which acceptance criteria must address. The major remaining question addressed is, given the difficulties, in what forms are acceptance criteria generated from the requirements? This subsection examines the following types of acceptance criteria:

1. Control objectives
2. Acceptance tests
3. Loss estimates

4. Formal verification

The former two types are in common use today. The latter two, while being used experimentally, are perhaps better viewed as potential types of criteria.

1. Control objectives. The term "control objectives" is used to refer to control hierarchies such as proposed in [CIC75] or [EAF80] involving structures of control objectives, standards, and (perhaps) requirements or techniques. These are basically security requirements phrased in terms of high-level objectives and lower-level functions. In many situations, qualitative aspects (e.g. performance, penetration resistance) are not addressed in the hierarchy. The argument may thus be made that these are not true acceptance criteria. They do represent, however, probably the most commonly used metric serving as acceptance criteria. In some areas, they may be the best criteria achievable. The philosophy behind using such "objectives" as acceptance criteria is that they establish the spirit and intent of the requirements against which evaluation judgments can be made.

   In using such an approach, the challenge would be not only to formulate the policy structure but also to apply it within the context of the entity to be evaluated. For example, in a simple situation this might be done using a matrix of required control functions versus "activities" (see [MAI76]) of the entity. Checklists (e.g. [AFI79], [EAF80,83]) could be useful in attaining this integration, preferably with several being analyzed and a new one (e.g. [DOA80], [HHS78]) formulated for the particular evaluation.

   Evaluation reviews to determine if such criteria are satisfied can take place at any time during development or operation. Reviews occurring during early development are usually referred to as "design review" evaluations. Their purpose is to determine whether the requirements are adequately embodied in the specifications. If this evaluation step includes formal approval of the specifications, subsequent evaluations (e.g. testing) would have to ensure compliance with the specifications as well as the initial requirements. In a sense, then, compliance with the specifications (which embody the requirements and acceptance criteria) would then also become an acceptance criterion. It would be a much more implementation-specific criterion, however.

The audit community in particular uses control objectives as a form of security guidance and acceptance criteria. These often include further guidance to evaluators to be used in assessing compliance with the objectives. The following example is from Control Objectives-1980 [EAF80, p. 103] [51].

> "3.6.2.1 On-line systems and programs should be designed to require user identification and verification.
>
> a. Determine the identification techniques used and the manner in which they are controlled.
>
> b. If terminal identification or physical address is in use, test the effectiveness of this control by execution of programs from other terminals.
>
> c. If passwords or user identification is in use, determne if the system will permit execution of programs without the entry of passwords.
>
> d. Review the table of passwords and constraints used to determine the extent to which users are controlled in the execution of programs and the execution of specific functions within the program.
>
> e. Review installation procedures for regular modification of passwords and determine if passwords are regularly modified to provide protection against the use of passwords by unauthorized persons who may gain knowledge of them."

Another example is Figure 4-8 (from Computer Audit Guidelines [CIC75], p. 286).

In tailoring general control objectives such as these for organization-specific use, they are often supplemented with required control functions or features (i.e. more detailed requirements). For example, from the HHS "Part 6" Manual [HHS78, Chap. 6-30, p. 21], "there must be programs that will clear all sensitive data from the system, or make it inaccessible." DoD security regulations also detail such required functional capabilities.

For example, DoD 5200.28-M [DOD79, p. 24] describes required hardware features such as "error detection should be performed on each fetch cycle of an instruction and its operant".

---

[51] [EAF83] treats this subject in Section Z4.

Control objectives and the requirements deriving from them provide good security guidance. They do not, however, reduce the almost completely judgmental nature of compliance evaluation. Absolute requirements such as the HHS quotation above still leave many ambiguities which would have to be resolved by judgment during evaluation. For example, with regard to the quotation, many questions can be asked:

1. When and how will the programs be used? Can they be interrupted during use?

2. What assurances must there be that the programs work properly? What is an acceptable measure of their quality?

3. What is really technically necessary to clear data from the system? DoD recognizes that, depending on the threats, a simple over-write may not be sufficient and as a result distinguishes between downgrading for reuse and declassifying for release [DOD79, pp. 30-34]. What would be needed here?

Of course judgment cannot be eliminated from the overall certification process. (See Section 1.2 for definition of certification and its connection to evaluation.) If it is not required in the evaluation, it must be applied in formulating the acceptance criteria. It is impossible, however, to confine all judgmental issues to the early definition of criteria. This is easily illustrated by the changes made to any initial design as a result of its implementation. The only resolution is to apply as much judgment as possible in the early definition of criteria, while recognizing that further judgment cannot be avoided in evaluating compliance.

The work being done on control objectives is important because it is at least forcing wide consideration of what objectives are reasonable and acceptable. This is a critical precursor to the problem of defining organization-specific security capabilities and acceptance criteria.

2. Acceptance tests. Most "true" acceptance criteria are based on acceptance tests. This is because they can provide a more readily measurable set of criteria (i.e. test success or failure) and because they assess the implemented (i.e. actual) system. As noted in Section 4.1.5.1.2, it is not a trivial process to ensure that pass/fail criteria are precise. Criteria must answer questions such as the following:

1. Under what conditions (e.g. system state or load)?

2. What constitutes success or failure (e.g. consider partial and extraneous results, intermittent success)?

Acceptably precise criteria, however, can be defined.

As can be inferred from Section 6.2.1, there are three primary forms of acceptance tests[52]: functional, performance, and penetration resistance. Acceptance criteria thus should be stated in terms of precise capabilities, performance (e.g. error rate, response time, throughput), and penetration resistance in accomplishing the capabilities.

A form of acceptance criteria of particular interest here is penetration resistance. Tests to evaluate this can, like other acceptance tests, be highly structured with detailed predefined procedures or loosely structured, perhaps involving only a time period set aside for testing. The evaluation might involve detailed internal analysis of the system, sometimes assisted by automated tools. Internal as well as external security boundaries might be examined. (Conceivably such testing may take place during development, along with program testing.)

Even with unstructured penetration tests, failures can be precisely defined. Acceptability criteria, however, are more difficult to define. There are several reasons for this. One (which applies for all acceptance criteria) is that it may not be clear how many individual test failures are required before the threshold of outright system rejection is reached. For example, if 50% of the security tests fail, this would clearly represent a pervasive lack of security which could not be repaired by simply modifying the system to pass the failed tests. On the other hand, a few failures will be inevitable, even in the best systems. (This possibility stresses the need for accept/reject review points throughout the development cycle, although criteria for rejection based on a faulty design must be highly judgmental in nature).

Another difficulty arises because there is no assurance that any existing system could withstand penetration attacks by knowledgeable persons. The use of penetration resistance as an acceptance criterion, then, has been difficult because the degree of protection actually required has not been technically attainable[53]. Even if the success of

---

[52] Development methodology is checked by documentation and observation, not by performing test on the system.

penetration is highly unlikely, as in some of the new DoD systems being developed primarily for security, precise criteria are still not readily defineable as shown in Figure 6-3[54]. Currently, rigorous criteria such as those in the figure would usually be more appropriate for applications allowing a restricted set of user capabilities than for the operating systems.

Despite these shortcomings, acceptance tests remain the best "true" acceptance criteria available today, because of their fairly ready demonstrability.

3. Loss estimates. Theoretically, risk assessments could be ideal vehicles for evaluation, based on acceptance criteria stated in terms of acceptable loss, whether quantified as dollars (e.g. ALE) or stated as risks (e.g. high, low). Acceptable losses could be broken down by component (e.g. application A, B; operating system) and categorized by type (e.g. disclosure, denial of service) or by areas of management interest (e.g. statutory loss, competitive disadvantage). Evaluation based on this framework would entail estimating how the varying quality of a control would affect such factors as threat frequency and the rate of successful attack.

---

[53] While penetration resistance has as a result seen limited use as an acceptance criterion, it might provide a testable basis for comparing two systems. For example, SDC performed penetration studies of two systems, counting the number of flaws found per man-month of effort in order to compare their penetration resistance [WEI80]. In one system the number of flaws found was large and continued to rise throughout the study. In the other system, the number of flaws found was much smaller and none were found after the early phases. The conclusion was that the latter system was much more penetration resistant. (While the studies were clearly not "scientific", they were both performed by the same penetration team).

[54] An interesting variation on acceptance criteria for penetration testing is noted by HHS [HHS78, Chapt. 6-30, p. 11]. The context is that of a controlled penetration study being conducted by systems security personnel in order to "dramatize to management the need for more effective safeguards."

> "At the end of a controlled penetration study, the evaluator must be able to demonstrate beyond the shadow of a doubt that any employee or outsider could have accomplished the same ends. The chance of success for this technique should be better than 90 percent, since a failed attempt will have a negative impact on an overall evaluation and on the credibility of the evaluator."

a. Externally exploitable disclosure violations. This is the class of error in which a user outside the security boundary can cause disclosure of information without the involvement of any malicious or instrumented, trusted or untrusted code, and without any maliciousness on the part of individuals within the security perimeter. This type of error will prevent certification and will have to be corrected before the system can be placed into operation.

b. Specification inconsistencies. This includes security errors which might only be internally exploitable (i.e. from untrusted software) but represent a security design which contradicts the approved specification. Flagrant externally exploitable data integrity or denial of service flaws would also be included here. This type of error will probably prevent certification and require correction. Each such error will be examined on a case-by-case basis.

c. Internal security flaws. This includes flaws which are not externally exploitable (except for minor data integrity or denial of service violations) and which do not contradict the specification. This type of error will probably not prevent certification or require correction before the system is placed into operation. Action must be taken, however, to correct the flaws as soon as possible. The right is reserved to make exceptions for unusual cases.

Figure 6-3. Sample Penetration Resistance Acceptability Criteria (Adapted from [NEU80].)

This might be a very useful form of acceptance criterion for installation evaluations (although the problem of obtaining reliable frequency and loss estimates would have to be resolved). In fact, the provisional Air Force AFRAMP considers an ALE of 10% or less of asset value to be acceptable, and defines this figure as their "Standard Baseline Risk Assumption" [AFRAMP, vol. I, para 9-4]. There is one primary difficulty with this process, however, for systems and applications (summarized in Section 5.2.2). Namely, the risk assessment mechanisms are not of sufficient precision to be used in evaluating detailed internal controls. The reason is that while threat frequencies and the rate of successful attack can be useful at a high level, they become increasingly meaningless and impossible to determine at lower levels. Even if attack frequencies could be determined at these lower levels, it would be extremely difficult to determine how these frequencies would be affected by changing the internal control posture, since penetrators could simply go around strengthened controls. Risk assessments do not readily consider such interdependencies.

4. Formal verification. The formal verifiability of a design is being used today in DoD as a system acceptance criterion. This is still primarily a research area, however. Ultimately, as noted in Section 4.1.6.10, formal verification has the potential to play an important role in certification policy.

6.2.4 Conclusions

While much progress remains to be made, the last few years have seen a growing awareness of the need for computer security. As a result, many organizations are now defining computer security requirements for the first time. The facet of these requirements which seems most commonly ignored is acceptance criteria. The role of acceptance criteria is to ensure that requirements include sufficient definition of:

- What degree of quality is required?
- What will be examined in evaluating the degree of quality?

Since acceptance criteria emphasize qualitative characteristics, one way to increase awareness of the need for acceptance criteria would be to structure security requirements (and perhaps evaluations) around qualitative classes. One such structure, proposed here, includes functional, performance, penetration resistance, and

methodological requirements.

Based on examination of these classes, an important finding of this technology assessment is that most existing evaluation methodologies are of very limited applicability in evaluating compliance with the qualitative (e.g. control performance and penetration resistance) aspects of acceptance criteria. More useful for these aspects would be situation-specific testing and design review. The primary areas of applicability of existing evaluation methodologies seem to be:

- Evaluating entities for which computer security requirements and acceptance criteria have not been well defined. (This would tend to make them more applicable to operational as opposed to developmental evaluations.)
- Evaluating for control existence (as opposed to quality).
- Defining functional requirements (as opposed to performance or penetration resistance requirements).

Four types of acceptance criteria were examined: control objectives, acceptance tests, loss estimates, and formal verification. Control objectives and the required control functions based on them serve as useful design guidelnes and also serve as good high-level acceptance criteria. Evaluations to determine compliance with these criteria would rely heavily on judgment. The DoD initiative to develop a highly-experienced, centralized evaluation group to evaluate systems based, essentially, on control objectives has interesting promise here. Acceptance tests serve as good low-level acceptance criteria. Greatly reduced amounts of judgment are required in determining compliance with acceptance tests. However, high amounts of judgment are required in formulating a sufficient set of acceptance tests.

Risk assessments do not have sufficient precision to define low-level criteria for the internal control of systems and applications, but may be of use in defining and measuring compliance with loss-estimate-based criteria for installatons. Formal verification remains somewhat experimental but has substantial future promise.

Each form of acceptance criterion has a use. There is no one single form which can suffice. Indeed, if the ability existed to precisely define and measure one form of criterion, the system would likely be optimized to meet that and might prove unacceptable by another, less measurable criterion. To defend against such optimization, DoD procurements often require many forms of acceptance

criteria. For example, some DoD procurements specify use of the following as acceptance criteria: control objectives and specific control functions (perhaps validated by risk assessments); formal verification; and acceptance tests (to measure functionality, performance, and penetration resistance).

Although predefined acceptance criteria can greatly reduce the judgment required in evaluation, the need for judgment remains. This judgment must not be forced prematurely. For example, if an entity is being developed or procured, many of its features and procedures may not be known initially. As a result, many desirable detailed acceptance criteria (e.g. specific acceptance tests) cannot be formulated at the outset. The development or procurement agreements, therefore, should reserve the right of the customer to apply judgment when sufficient information becomes available. For example, where quantitative performance requirements exist, there must be agreement that the requirements be applicable and demonstrable under customer-approved configurations and conditions. As another example, judgment can be reserved via the requirement for customer approval of designs and proposed tests. With respect to testing, it would, of course, be preferable to utilize an independent test team, ideally without the developers even knowing which tests were to be performed.

Overall, since acceptance criteria serve to guide design as well as evaluation activities, the objective is to apply as much judgment as possible in the early definition of criteria. This should both improve the quality of systems and reduce the amount of judgment required in their evaluation.

CHAPTER 7

DOCUMENT OVERVIEW

Each section of this document is summarized below. Before proceeding with this summary, however, it is important, for perspective, to emphasize the key underlying problem.

The basic underlying problem is the widespread lack of security awareness. This in turn accentuates the need for security evaluations which permit certifications of sensitive applications, systems, and installations for their security posture. The 'hacker' incidents from 1983 onward have produced a much needed stimulus in this area, but it remains to be seen what lasting effect these have on the awareness problem. The security problem includes not only such difficult issues as accommodating the inherent penetrability of complex systems, but also more manageable issues such as anticipating disasters and placing limits on individual trust. Unfortunately the increased and improved use of security evaluation methodologies will not solve this problem. The only solution is to instill security awareness among those who design, develop, operate, maintain, and use systems and, most important, those who manage organizations. The emphasis (in resources and attention) placed on security by top management remains the most important factor in ensuring optimal and acceptable security. The motivational spirit and justification behind the emphasis must also be conveyed in order to motivate conscientious performance (i.e. management must convince employees why security justifies the resources and attention). So, while the issues and methodologies summarized below are important, their optimal resolution and use remain contingent on the successful treatment of this underlying "cause".

7.1 INTRODUCTION

This technology assessment is a summary and assessment of an investigation of methods for measuring the level of computer security. This effort by NBS is part of its Computer Security, Integrity, and Risk Management Standards Program and is in response to its responsibilities assigned under the Brooks Act (PL96-306) and mandated by OMB Circular A-71, TM1. Th Introduction chapter defines seventeen heavily used terms and then describes the content of the document as

well as the sources of the information and the approach to developing the report. Before the evaluation approaches were discussed in the body of the document, it was felt that a discussion of environmental factors and control issues were essential for a deeper understanding. Chapters 2 and 3 treat these areas respectively. Chapter 4 describes the twenty five approaches that were reviewed, Chapter 5 discusses the state of the art, and Chapter 6 discusses the impact of security policy.

7.2 ENVIRONMENTS

Environmental factors include everything that influences a system. This assessment examines environmental factors which influence both system security requirements and security evaluation requirements. Some factors are more important than others. In the area of functionality, for example, user capability and degree of sharing are the primary characteristics which determine the level of protection requirements. This document proposes a structured categorization of such environmental influences (see Table 2-3). A potentially fruitful area needing research is that of determining the extent to which a small number of environmental influences can be used to define and categorize systems for security purposes. Environmental influences greatly determine the nature of an organization-specific evaluation methodology.

7.3 CONTROLS

7.3.1 Control Groupings Or Structures

Many alternative ways to classify and group controls were examined. Advantages and disadvantages of different groupings or structures were analyzed. The selection of one or more control grouping(s) influences all phases of a security evaluation. Consideration of control structures is thus critical to the effective, efficient performance of a security evaluation. It would be dangerous to blindly adopt an arbitrary general structure. Alternative groupings must be thoroughly analyzed and an organization and situation-specific structure adopted to meet the needs of a particular evaluation. The major roles served by control groupings in the security evaluation process are to:

1. Support or provide overall partitioning of the evaluation analysis.
2. Support the objectives of the evaluation.
3. Support the philosophy or approach of the evaluation analysis.
4. Permit focusing the skills of technical specialists.

5. Correlate with organizational responsibilities, structures, policies, and documents.
6. Prioritize analysis.
7. Clarify the purpose of controls.
8. Clarify the overall control problem.

7.3.2 Exposure Groupings Or Structures

In analyzing control groupings, it became apparent that exposure groupings or structures are also of major significance in security evaluation since they define the underlying problem being addressed by controls. There are two distinct groupings for exposures (i.e. impacts resulting from loss). These are "mutually exclusive" (to the first order) and "overlapping". Mutually exclusive exposure groupings are typically used in risk assessments and overlapping exposure groupings in audits. Overlapping exposures may be much more meaningful to management because of the type questions asked in this kind of analysis. Since evaluations are done for management, it is critical to report findings in terms that management can best understand and use. Improper definition of exposure structures, however, would be a likely pitfall in the use of established security, risk assessment, or audit methodologies for purposes other than those originally intended. Exposure groupings, then, like control groupings, are very important in tailoring a security evaluation process to meet management objectives.

7.4 EVALUATION METHODOLOGIES/APPROACHES

Twenty five methodologies/approaches from the security, audit, and risk assessment communities were examined. Based on this examination, the conclusion was drawn that there is no widely accepted existing way to measure a level of computer security. This derives from the lack of generally acceptable "levels" and the lack of precise "measurement" capabilities. All top security experts consulted were pessimistic about near-term breakthroughs. Existing methodologies are useful, however, in guiding and structuring security evaluation and can produce meaningful results (or misleading results if improperly used). Following are major features of the individual methodologies, along with some general conclusions.

1. Touche Ross & Co. [MAI76] This audit methodology has been in use for almost thirteen years. At least 50-75 thousand copies have been distributed. The methodology may have hundreds of users. The book describing the methodology is an excellent and comprehensive tutorial. A book of case studies is also available from the firm. Matrices are used for analysis. There is no checklist of questions.

2. Peat Marwick Mitchell & Co. [PMM80] The Data Processing Security Evaluation Guideline (DPSE) is a proprietary audit methodology and has been in use and under continuous development since 1973. It has been used for hundreds of clients. Ten security checklists are included. A final numeric "score" is produced. Weighting of the ten security areas is done by a user group using the Delphi method.

3. SRI International/USC Information Sciences Institute(ISI). [NEM78] The primary distinguishing feature of this security approach (it is not a methodology since it is not complete) is its use of an ISI-developed protection flaw categorization. The categorization describes and provides symptoms of common design and implementation errors which create vulnerabilities. The approach also emphasizes the consideration of development methodology.

4. Department of Defense. [DOD83] As part of the DoD Computer Security Initiative, DoD has established a laboratory evaluation process of industry-developed systems, that is resulting in an evaluated products list. MITRE originally developed a set of protection levels to serve as evaluation criteria and DoD added to these. The approach centralizes evaluation of the most complex components, permitting this task to be done by highly-qualified specialists.

5. Testing. Testing has long been the primary method of evaluating security. There are two categories - external and internal. Both can include security testing. External security testing should include independently-defined penetration tests and precise pass/fail criteria. Internal security testing is beginning to avail itself of tools such as assertion checkers and path flow analyzers. Two research areas of relevance to internal security testing are measures of coverage and software quality metrics. Both may permit quantitative measurement of freedom from errors. Testing will remain crucial for security evaluation.

6. Canadian Institute of Chartered Accountants. [CIC75] Computer Audit Guidelines contains a detailed, qualitative, structured control evaluation methodology. It assesses whether the organization-specific implementation of a set of generic control techniques meets minimum standards. It incorporates checklists.

7. Arthur Anderson & Co. [AAC78] Arthur Anderson's document has gained prominence both for its discussion of control objectives and its presentation of a transaction flow review approach to evaluation. A proprietary adaptation of the approach is being used for security reviews.

8. AFIPS Security Checklist. [AFI79] This checklist presents an excellent basis for a detailed security evaluation. Structure, comprehensiveness, and overall quality are outstanding. It may be that no single security checklist can compete with this one for either completeness or clarity.

9. Internal Controls for Computerized Systems. [FIT78] This document contains a checklist of over 650 controls organized into nine control groups. Each control group has a matrix with concerns/exposures along one axis, resources/assets along the other axis, and appropriate controls in each box of the matrix. This document does not present a security evaluation methodology, but rather a useful tool to use within a methodology.

10. Coopers & Lybrand. [C&L82] [HAL85] Coopers & Lybrand has a proprietary integrated approach and methodology of long standing. There are nine major steps to one of their reviews with evaluation of internal controls and functional testing of controls being two of them. A control questionnaire and a matrix of responses is used. This activity is under the jurisdiction of the C & L Computer Assistance Group (CAAG), which also engages in extensive training.

11. Auditing Computer Systems [PER81] Although the overall document is concerned with the general practice of auditing, Chapter 5 on "Auditing an Application" deals with how to review for internal controls -- a subject very relevant to security evaluation. The approach uses an internal controls checklist and a control evaluation worksheet on which controls are numerically rated. Applications are then rated by arithmetic mean. This offers the user a first order type quantitative evaluation.

12. Information Security Handbook. [WIL80] This brief handbook addresses establishing and auditing information security. It focuses on preventing information being disclosed to an unauthorized recipient. It contains twenty checklists, each of which addresses an important area of concern. It gives little information on implementation.

13. Department of Health and Human Services. [HHS78] [HHS82] The 1978 HHS ADP Systems Security Manual was oriented around checklists used for security evaluation. The manual defined the HHS security program, principles, responsibilities, and authorities. It also included risk management guidelines and provided security requirements checklists in eight areas of concern.

    The revised 1982 version of the manual is oriented around a matrix of minimum security requirements and safeguards. Managers of computer systems are expected to incorporate the minimum safeguards in facilities and applications until a security review reveals the need for

more specific controls. Detailed guidance is given in eight areas.

14. Department of Agriculture. [DOA80] [DOA84] In 1980 USDA was developing a security evaluation methodology for compliance with OMB A-71, TM1. It had two parts, internally-developed questionnaires and contractor-developed methodological guidance. Emphasis was placed on interview techniques. The methodology was to be tested within DoA but funding for performance of the evaluations was a problem.

    In order to avoid the problems of personnel and cost foreseen in using this centralized approach to evaluation, USDA rewrote their security standards in 1984 and added a security policy document that puts responsibility for carrying out the security program in the various USDA agencies. This new document refers agencies to [FIP65] for risk analysis, [FIP102] for certification, and [GAO81-1] for evaluation questionnaires.

15. GAO Audit Guides. [GAO81-1] [GAO81-2] [GAO81-1] advises generalist auditors on review steps to take when using computer produced data in their evaluations. This document's suggestions for determining reliability of such data are (1)confirmation with independent sources, (2)reasonableness of data, (3)comparison with independent sources, and (4)a User Satisfaction Questionnaire.

    [GAO81-2] provides detailed guidance for information system evaluations that result from recommendations of such an evaluation or from other audit requirements in GAO. This document looks for system or application reliability in processing data in a timely, accurate, and complete manner. Extensive checklists and questionnaires are used in a four phase evaluation. Controls are grouped into top management, general, and application controls.

16. Department of Energy. [DOE83] [DOE84] [DOE84] is the main statement of DOE's approach to computer security and internal control. The document compares ADP security with ADP internal control and draws the important conclusion that security controls are a proper subset of internal controls. Detailed questionnaires are given for performing vulnerability assessments. A scoring scheme is included which is heuristic. The latter half of the document discusses internal controls techniques for management, operations, and applications.

    [DOE83] gives guidance on planning and performaing internal controls reviews. Eight major activities for performing the reviews are discussed in detail. Evaluations are judgmental.

17. Formal Verification. Design and program verification have promise and are applicable to security evaluation. Existing methodologies are experimental, i.e., Gypsy, Hierarchical

Development Methodology, Formal Development Methodology, and Stanford Verifier. The document discusses pertinent issues such as cost, formal specification difficulties, and possible inapplicability to systems that are changing rapidly.

18. FIPS PUB 65. [FIP65] This high-level methodology has evolved over twelve years and informs the Federal ADP Manager how to do a risk assessment. The fundamental operand is the set of assets. It serves as a high-level statement of purposes and methods and, when applied, produces an order-of-magnitude Annual Loss Expectancy [ALE] in dollars.

19. Air Force Risk Analysis Management Program. [AFRAMP] This is a highly detailed and highly structured risk assessment methodology. It provides extensive guidance in evaluating assets and estimating threat frequencies and magnitudes. It has been shelved in favor of detailed questionnaires.

20. Department of Agriculture. [DOA77] This handbook was developed to assess the current security position, to raise security awareness, and to cost-effectively allocate resources. It has some similarity to [FIP65] but emphasizes threat rather than asset. It has only two classes of risk -- major and minor. It actively involves users and calls for a team approach. It is currently superseded by use of [FIP65] in USDA.

21. SDC Navy Risk Assessment Methodology. [SDC79] This highly structured methodology allows assets to be evaluated using either dollars or a qualitative non-dollar value technique and incorporates the explicit treatment of vulnerabilities. It also includes forms and checklists.

22. Risk Analysis and Management Program. [IST79,81] IST/RAMP is an automated method and tool for obtaining quantitative estimates of expected losses caused by threats at data processing facilities with multiple applications. It runs on IBM DOS and OS systems. It modifies the [FIP65] loss expectancy equation with appropriate heuristic interaction functions. It can be used to optimize back-up plans; to obtain single occurrence losses; to optimize safeguard selection; and for several other purposes.

23. Relative Impact Measure (RIM). [NIE80] SRI International is developing this technique to measure the relative impact on an organization of vulnerabilities of its computer system's integrity. It is easy and inexpensive but does not provide a measure of monetary impact.

24. Fuzzy Risk Analysis. [HOF80] This risk analysis approach is based on earlier work using fuzzy set theory. It uses linguistic terms rather than numbers to avoid giving users unwarrented confidence in results. Numerical estimates are

not required. It is currently not being actively pursued.

25. Security Assessment Questionnaire. [IBM80] IBM85] This questionnaire contains fourteen categories that are divided into three key security areas. Yes/No answers are expected. Each category is given a judgemental risk rating by the reviewer with high risk areas implying a need for corrective action. There is no guidance on the risk rating. The questionnaire may be suitable for a high level basic evaluation (See [FIP102]). The updated version of this questionnaire, [IBM85], divides its categories into four key security areas, rather than three.

7.5 SUMMARY OF THE STATE-OF-THE-ART

7.5.1 Similarities And Differences

Findings are that the methodologies have a fairly consistent general underlying structure, and they all fail to mention major functions and interrelationships. Little empirical data exists on the use, success, or failure of the methodologies. All methodologies differ in their specific approaches.

Audits and risk assessments differ from each other and from pure "security" analyses. The differences derive from their different purposes and are summarized in Table 4-14. These differences influence the nature of the evaluation process. Also influencing the nature of the evaluation process are the differing evaluation objectives. Evaluations may determine flaw susceptibility or focus on detecting actual flaws. They may detect flaws associated with anticipated or unanticipated threats or attacks. Finally, evaluations may assess proper operation or acceptable development.

Risk assessment methodologies support general determination of flaw susceptibility and can be used to detect flaws associated with anticipated threats or attacks, especially for installations. Audit methodologies support detailed flaw susceptibility determination and fairly detailed detection of flaws associated with anticipated threats or attacks, especially for applications. Security methodologies support the detection of both types of flaws. Analysis for flaws associated with unanticipated threats or attacks is often referred to as penetration analysis. While operational evaluations are required to show that defenses are being maintained, developmental evaluations are becoming increasingly common, due to the difficulty of retrofitting security controls.

7.5.2 Approaches For Structuring Analysis

Four common approaches for structuring evaluation analysis are use of matrices, use of checklists, transaction flow analysis, and a loosely structured approach. Uses, advantages, and disadvantages of each are discussed. Matrices help to structure the system and process and help in examining interrelationships but their use is cumbersome and time-consuming. Checklists help to ensure completeness, capture complexity, and heighten awareness but their use is also cumbersome and time-consuming. The transaction flow analysis focuses attention and improves perspective. Loosely structured approaches are fast and simple but provide little documentation and require highly experienced personnel. All of these approaches permit different mechanisms for focusing attention.

7.5.3 General Evaluation Issues

Many issues associated with security evaluation are mentioned such as the need to better define underlying assumptions. Three issues discussed at length are quantification, uncertainty and bias, and integration with the decision process. The three discussions are summarized here.

Both absolute and relative quantification are misleading with the dangers proportional to the complexity of the element being measured and the extent of supporting data. The challenge with using supporting data is to assess, accommodate, and reflect its quality. Absolute quantification in risk assessment is particularly vulnerable to error. Quantification is a useful but a very volatile tool. It can be used to forcefully promote findings but can also serve to camouflage faulty assumptions or analysis.

Uncertainty and bias have received almost no attention in security evaluation literature. Inaccuracies deriving from uncertainty and bias are strongly relevant to security evaluation. Techniques exist to anticipate and offset bias. Estimates based on uncertainties should include probability distributions. Complicating this is the fact that individuals can not reliably state the confidence of their own estimates. The Delphi method has applicability here. The field of decision theory in general has significant applicability to security evaluation and certification.

The issue of integrating evaluations with the decision-making process refers to the fact that decisions involving security risks must take place within the same organizational framework as other decisions involving other forms of risk. This can strongly influence both the content of evaluations and the presentation of findings. Evaluations must be keyed around information that management needs.

understands, and most importantly, uses.

7.5.4 Conclusions

There is no widely accepted existing way to measure a level of computer security. This derives from both the lack of generally accepted "levels" and the lack of precise "measurement" capabilities. There is a strong need for guidance in security evaluation, however. This conclusion stems not so much from shortcomings in existing methodologies as from the much more widespread lack of evaluation programs and capabilities in government and industry. In the process of formulating this guidance, there is much to be learned from the general approaches and experience of the audit and risk assessment communities.

Existing evaluation methodologies all have advantages and disadvantages with different ones being preferable for different people and situations. In a single evaluation, it will often be desirable to use several different approaches in parallel for different organizational or system components. It will also be desirable to combine approaches to form hybrids. Evaluations tend to be much more practical during development than during operation.

The most critical need in performing a security evaluation is the use of people who have sufficient motivation, intelligence, security perspective, and knowledge of the entity being evaluated to perform the work. Methodologies can help provide training and perspective and can guide the work, but people must still do the work. No methodology, no matter how detailed, can supplant the need for judgment, common sense, and hard work. Indeed, the use of a detailed methodology can easily impede evaluation if it diverts attention from the basic analysis at hand. For some evaluations, a sufficient "methodology" may essentially be to review user manuals with common penetration approaches in mind and run tests in likely problem areas. With any methodology the most important need is to adapt it to meet management, organization, and situation-specific characteristics, resources, needs, and objectives. The method must not obscure the mission.

7.6 SECURITY POLICY IMPACT

7.6.1 Sensitivity Distinctions

Sensitivity distinctions are often made up of two generic structural types: horizontal and vertical. While horizontal structures would theoretically suffice, both are usually used together. Vertical structures are often used to define levels of

protection (as opposed to sensitivity) requirements. Categorization of data can be very difficult. Attempts at such categorizations have been made by both civilian agencies and private sector organizations. There does not seem to be a strong correlation between security evaluation and the organizational approach taken for sensitivity distinction.

7.6.2 Acceptance Criteria

Acceptance criteria are specialized security requirements which respond to the question "how will we decide if the product is acceptable?" Their role is to ensure that the requirements include sufficient definition of:

- What degree of quality (e.g. performance, penetration resistance) is required?
- What will be examined in evaluating the degree of quality?

Acceptance criteria thus should be "measurable" or demonstrable features of required security functions which characterize their desired quality.

Definition of acceptance criteria is the most important and the most difficult task of security evaluation and certification. It is the most important because it underlies and often defines both the design of a system and the associated evaluation and certification process. It is the most difficult because the process of defining acceptance criteria is subjective, complex, dynamic, and based on little experience. Each of these is expanded upon. For example, in illustrating subjectivity it was noted that, since absolute security is not achievable, it is not meaningful to say "the system must prevent data disclosures" when it is impossible, even with unlimited resources, to absolutely assure this prevention.

Classes of requirements and corresponding evaluations were defined. The classes are important, because they emphasize the different knowledge and skills associated with both defining the different types of requirements and performing the corresponding evaluations. The classes are: functional, performance, penetration resistance, and methodological. The applicability of different evaluation methodologies examined in Chapter 4 is discussed for each class.

7.6.3 Conclusions

In general, most existing methodologies are of very limited applicability in evaluating compliance with the qualitative (e.g.

control performance and penetration resistance) aspects of acceptance criteria. More useful for these aspects would be situation-specific testing and design review. The primary areas of applicability of existing evaluation methodologies seem to be:

- Evaluating entities for which computer security requirements and acceptance criteria have not been well defined. (This would tend to make them more applicable to operational as opposed to developmental evaluations.)
- Evaluating for control existence (as opposed to quality).
- Defining functional requirements (as opposed to performance or penetration resistance requirements).

Four types of acceptance criteria are discussed: control objectives, acceptance tests, loss estimates, and formal verification. The former two types are in common use today. The term control objectives refers to control hierarchies such as those involving groupings of control objectives, standards, and (perhaps) requirements or techniques. Although often only functionally oriented (giving little qualitative guidance), these are probably the most commonly used "metric" serving as acceptance criteria, and in some areas may be the best criteria available. They serve as useful design guidelines and establish the spirit and intent of the requirements against which evaluation judgments can be made. Evaluations to determine compliance with these criteria rely heavily on judgment. The DoD initiative to develop a highly-experienced, centralized evaluation group to evaluate systems based, essentially, on control objectives has interesting promise here. Acceptance tests serve as good low-level acceptance criteria. Greatly reduced amounts of judgment are required in determining compliance with acceptance tests. However, high amounts of judgment are required in formulating a sufficient set of acceptance tests. Risk assessments do not have sufficient precision to define low-level criteria for the internal control of systems and applications, but may be of use in defining and measuring compliance with loss-estimate-based criteria for installations. Formal verification remains somewhat experimental but has future promise.

Each form of acceptance criterion has a use. There is no one single form which can suffice. If the ability existed to precisely define and measure one form of criterion, the system would likely be optimized to meet that and might prove unacceptable by another, less measurable criterion.

Since acceptance criteria serve to guide design as well as evaluation activities, the objective is to apply as much judgment as possible in the early definition of criteria. This should both improve the quality of systems and reduce the amount of judgment required in their evaluation.

APPENDIX A

REFERENCES

AAC78 A Guide for Studying and Evaluating Internal Accounting Controls, Arthur Andersen & Co., January 1978.

ACM79 "ACM Forum; Comments on Social Processes and Proofs", Communications of the ACM, Vol. 22, No. 11, November 1979.

ADA78 Adams, Donald, Book Review: Auditing Computer Systems, The EDP Audit, Control, and Security Newsletter (EDPACS), Vol. VI, No. 3, September 1978.

ADR81 Adrion, W. Richards, Martha A. Branstad, John C. Cherniavsky, Validation, Verification, And Testing of Computer Software, NBS Special Publication 500-75, April 1980.

AF79 1979 Summer Study on Air Force Computer Security, 18 June to 13 July 1979, Cambridge Massachusetts; The Charles Stark Draper Laboratory, Inc.

AFI74 AFIPS System Review Manual on Security, AFIPS Press, 1974.

AFI79 Security: Checklist for Computer Center Self-Audits, Peter S. Browne, AFIPS Press, 1979.

AFRAMP Risk Analysis Management Program, AF Reg 300-XX, Vol. I-III, no date.

AIC73 Auditing Standards Established by the GAO - Their Meaning and Significance for CPAs, American Institute of Certified Public Accountants (AICPA), 1973.

AIC78 Audit Considerations in Electronic Funds Transfer Systems, AICPA, 1978.

AIC79 Report of the Special Advisory Committee on Internal Accounting Control, AICPA, 1979.

AMR71 AMR's Guide to Computer and Software Security, AMR International,

Inc., 1971.

ANG71 Angoff, William H. (Ed), The College Board Admissions Testing Program: A technical report on research and development activities relating to the Scholastic Aptitude Test and Achievement Tests, College Entrance Examination Board, 1971.

APP83 Applebaum, C. H., J. Keeton-Williams, PVS-Design for a Practical Verification System, MITRE Technical Report MTR 8936, May 1983.

ASC78 Ascher, William, Forecasting, An Appraisal for Policy-Makers and Planners, The Johns Hopkins University Press, 1978.

BEL74 Bell, D. E., E. L. Burke, A Software Validation Technique for Certification: The Methodology, MITRE Technical Report MTR-2932, 12 November 1974.

BIG80 Biggs, Charles L., Evan G. Burks, William Atkins, Managing the Systems Development Process, Touche Ross & Co., Prentice-Hall Inc., 1980.

BIS78 Bisbey, Richard, Dennis Hollingworth, Protection Analysis: Final Report, University of Southern California/Information Sciences Institute (USC/ISI), ISI/SR-78-13, May 1978.

BOU83 Bound, William A. J., Dennis R. Ruth, "Making Risk Analysis a Useful Management Tool with Microcomputer Electronic Worksheet Packages," North-Holland Publishers, Computers and Security 2 (1983).

BOW78 Bowen, John B., "Are Current Approaches Sufficient for Measuring Software Quality?", Proceedings of the Software Quality and Assurance Workshop, November 1978.

BRA78 Bratman, Harvey, Guy Conner, Leah Danberg, Marcia Finfer, Joseph Yott, Certifiable Software Development Methodology, Prepared for Defense Communications Agency Command and Control Technical Center, System Development Corporation TM-6145/000/00, 30 June 1978.

BRA80 Braithwaite, Timothy, Security Controls Selection Procedure, private communication on a manuscript, 1980.

C&L82 This section is based on a private communication from Glenn C. Davis, partner in Coopers & Lybrand, New York, dated March 30, 1982.

CAM79 Campbell, Robert P., Gerald A. Sands, "A Modular Approach to Computer Security Risk Management", National Computer Conference Proceedings, AFIPS Press, 1979.

CAM80 Campbell, Robert P., A Guide to Automated Systems Security, Advanced Information Management Incorporated, 1980.

CAR78 Carlstedt, Jim, Protection Errors in Operating Systems: Serialization, USC/ISI, ISI/SR-78-9, April 1978.

CAV78 Cavano, Joseph P., and James A. McCall, "A Framework for the Measurement of Software Quality", Proceedings of the Software Quality and Assurance Workshop, November 1978.

CCT78 Computer Security Seminar, Sponsored by the Command and Control Technical Center, Defense Communications Agency, 27 January 1978.

CEQ76 A Recommended Air Pollution Index, Prepared by the Federal Interagency Task Force on Air Quality Indicators: CEQ, EPA, DOC, Issued by the Council on Environmental Quality, September 1976.

CHE80 Cheheyl, M. H., M. Gasser, G. A. Huff, J. K. Millen, Secure System Specification and Verification: Survey of Methodologies, MITRE Technical Report MTR-3904, 20 February 1980.

CIC70 Computer Control Guidelines, The Canadian Institute of Chartered Accountants, 1970.

CIC75 Computer Audit Guidelines, The Canadian Institute of Chartered Accountants, 1975.

COU74 Courtney, Robert H., "A Systematic Approach to Data Security", presented to the NBS Symposium on Privacy and Security in Computer Systems, 4-5 March 1974.

DAV80 Davis, Keagle W., Touche Ross & Co., private communication.

DCP78 Disaster Planning Guide for Business and Industry, Defense Civil Preparedness Agency, CPG 2-5, July 1978.

DEM79 DeMillo, Richard A., Richard J. Lipton, and Alan J. Perlis, "Social Processes and Proofs of Theorems and Programs", Communications of the ACM, Vol 22, No. 5, May 1979.

DEN78 Denning, Dorothy E., "Are Statistical Data Bases Secure?" National Computer Conference Proceedings, AFIPS Press, 1978.

DIN79 Dinneen, Gerald, P., "Computer Security Requirements in the DoD", Proceedings of the Seminar on the DoD Computer Security Initiative Program, July 17-18 1979.

DOA77 ADP Security Handbook (USDA DIPS Manual Supplement), United States Department of Agriculture, August 15, 1977.

DOA80 Application Review Self Audit, Department of Agriculture, in preparation, 1980.

DOA84 ADP Security Manual, United States Department of Agriculture, Office of Information Resources Management, DM 3140-1, July 19, 1984.

REFERENCES

DOD78 Security Requirements for Automatic Data Processing (ADP) Systems, DoD Directive 5200.28, 29 April 1978.

DOD79 ADP Security Manual, Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource-Sharing ADP Systems, DoD 5200.28-M, 25 June 1979.

DOD83 Department of Defense Trusted Computer System Evaluation Criteria, DoD Computer Security Center, CSC-STD-001-83, August 15, 1983.

DOE83 Guide for Performing Internal Control Reviews, DOE/M0104 Draft, April 1983.

DOE84 ADP Internal Control Guide, U. S. Department of Energy, DOE/MA-0165, August 1984.

EAF80 Control Objectives-1980, EDP Auditors Foundation for Education and Research, 1980.

EAF83 Control Objectives-1983, EDP Auditors Foundation for Education and Research, 1983.

EDP80 "Special Report: Computer Fraud and Embezzlement", Prepared by the EDP Analyzer Staff, Published in the Computer Security Manual, Computer Security Institute, 1980.

EPP80 Epperly, Eugene V., "The Department of Defense Computer Security Initiative Program and Current and Future Computer Security Policies," Proceedings of the Second Seminar on the DoD Computer Security Initiative Program, January 15-17, 1980.

FIA82 Federal Managers Financial Integrity Act of 1982, Public Law 97-255, September 8, 1982.

FIP31 Guidelines for ADP Physical Security and Risk Management, NBS FIPS PUB 31, 1974.

FIP39 Glossary for Computer Systems Security, FIPS PUB 39, February 15, 1976.

FIP65 Guidelines for Automatic Data Processing Risk Analysis, FIPS PUB 65, August 1979.

FIP101 Guideline for Lifecycle Validation, Verification, and Testing, FIPS PUB 101, June 6, 1983.

FIP102 Guideline for Computer Security Certification and Accreditation, FIPS PUB 102, September 27, 1983.

FIT78 FitzGerald, Jerry, Internal Controls for Computerized Systems, Jerry FitzGerald & Associates, 1978.

FIT81 FitzGerald, Jerry, Designing Controls into Computerized Systems, Jerry FitzGerald & Associates, 1981.

GAO79 Automated Systems Security - Federal Agencies Should Strengthen Safeguards Over Personal and Other Sensitive Data, U.S. General Accounting Office (GAO), LCD-78-123, January 23, 1979.

GAO80 Internal Controls Course, Outline, GAO, March 1980.

GAO81-1 Assessing Reliability of Computer Output - Audit Guide, General Accounting Office, AFMD-81-91, June 1981.

GAO81-2 Evaluating Internal Controls in Computer-Based Systems, General Accounting Office, AFMD-81-76, June 1981.

GER76 Gerhart, Susan L., Lawrence Yelowitz, "Observations of Fallibility in Applications of Modern Programming Methodologies", IEEE Transactions on Software Engineering, Vol. SE-2, No. 3, September 1976.

GER80 Gerhart, Susan, Presentation on AFFIRM, A Specification and Verification System, at the Third Seminar on the DoD Computer Security Initiative, 20 November 1980.

GIL80 Gilligan, John, Concept Paper on WWMCCS ADP Security, prepared for Defense Communications Agency, SDC Integrated Services, Inc., ISI-61/000/00, 2 May 1980.

GLA77 Glaseman, S., R. Turn, R. S. Gaines, "Problem Areas in Computer Security Assessment", National Computer Conference Proceedings, AFIPS Press, 1977.

GUD80 Gudes, Ehud, "The Design of a Cryptography Based Secure File System", IEEE Transactions on Software Engineering, Vol. SE-6, No. 5, April 1980.

HAL85 Halper, Stanley D., Glenn C. Davis, P. Jarleth O'Neill-Dunne, Pamela R. Pfau, Coopers & Lybrand, Handbook of EDP Auditing, Warren, Gorham, & Lamont, Boston and New York, 1985.

HEN80 Henderson, John C., Paul C. Nutt, "The Influence of Decision Style on Decision Making Behavior", Management Science, Vol. 26, No. 4, April 1980.

HHS78 Part 6, ADP Systems Security, Department of Health and Human Services (HHS) ADP Systems Manual, 14 September 1978.

HHS82 Part 6, ADP Systems Security, HHS ADP Systems Manual, July 2, 1982.

HOF77 Hoffman, L. J., Modern Methods for Computer Security and Privacy, Prentice-Hall Inc., Englewood Cliffs, N. J., 1977.

HOF80 Hoffman, L. J. and L. A. Neitzel, "Inexact Analysis of Risk", Proceedings of the 1980 IEEE International Conference on Cybernetics and Society, October 1980.

REFERENCES

HOH80 Hohenemser, Christoph, Kasperson, Roger E., Kates, Robert W., "Technological Risk: A Framework for Policy Formulation", presented at the annual convention of the American Association

HOL74 Hollingworth, D., S. Glaseman, M. Hopwood, "Security Test and Evaluation Tools: An Approach to Operating System Security Analysis", P-5298, The Rand Corporation, September 1974.

HOW77 Howden, William, E., "Reliability of Symbolic Evaluation", Proceedings of Computer Software and Applications Conference 1977, November 1977.

HUB79 Hubbert, James F., "Audit Criticality Measurement for EDP Applications", EDPACS, July 1979.

IBM80 Security Assessment Questionnaire, International Business Machines, GX20-2381-0, 1980.

IBM85 Security Assessment Questionnaire, International Business Machines, GX20-2381-1, 1985.

IIA77-1 Systems Auditability & Control; Audit Practices, The Institute of Internal Auditors, 1977.

IIA77-2 Systems Auditability & Control; Control Practices, The Institute of Internal Auditors, 1977.

IST79 RAMP, What It Is . . . How To Use It . . . What It Does . . ., International Security Technology, Inc., 1979.

IST81 Quantitative Risk Analysis: A Practical EDP Security Management Tool, Robert V. Jacobson, International Security Technology Inc., ISTI/81-04, 1981.

JAC80 Jacks, Edwin L., "Computer Security Interest in the Private Sector", Proceedings of the Second Seminar on the DoD Computer Security Initiative Program, January 15-17, 1980.

KRA79 Krauss, Leonard I., Aileen MacGahan, Computer Fraud and Countermeasures, Prentice Hall, Inc., 1979.

LAC74 Lackey, R. D., "Penetration of Computer Systems - An Overview", Honeywell Computer Journal, Vol. 8, No. 2, pp 81-85, September 1974.

LAM73 Lampson, Butler W., "A Note on the Confinement Problem", Communications of the ACM, October 1973.

LAU76 Lauer, H. C., On the Development of Secure Software, DCA/CCTC, SDC Report Number TM-WD-7826/000/01, 22 December 1976.

LID75 Linde, Richard R., "Operating System Penetration", National Computer Conference Proceedings, AFIPS Press, 1975.

LIN75 Linstone, Harold A., "Eight Basic Pitfalls: A Checklist", The Delphi Method, Techniques and Applications, Addison-Wesley Publishing Company, 1975.

LIP74 Lipner, Steven B., "Security Considerations in Information System Design", Proceedings of Approaches to Privacy and Security in Computer Systems, NBS SP 404, March 4-5, 1974.

LOB80 Lobel, Jerome, "New Audit Standards and Guidelines Needed for Distributed Systems", TeleSystems Journal (special issue on computer security), OnLine Software International, March-June 1980.

MAI76 Mair, William C., Donald R. Wood, Keagle W. Davis, Computer Control & Audit, The Institute of Internal Auditors, 1976.

MAN80 Mandell, Dick, System Development Corporation, private communication.

MAR73 Martin, James, Security, Accuracy, and Privacy in Computer Systems, Prentice-Hall, Inc., 1973.

MED79 Mednick, Robert, "Transaction Flow Auditing", Financial Executive, July 1979.

MIG73 Migliaccio, Guy R., "Computer Risk Insurance", Computer Security Handbook, Macmillan Information, 1973.

MIG80 Migliaccio, Guy, Marsh & McLennan (EDP Insurance Brokers), private communication.

MIL77 Miller, Edward F., Jr., Tutorial, Program Testing Techniques, Computer Software and Applications Conference 1977, Software Research Associates, 1977.

MIL78-1 Miller, Edward F., JR., "Program Testing", Computer, April 1978.

MIL78-2 Miller, Edward F., Jr., Program Testing Technology in the 80s, Software Research Associates, 1978.

MIN80 Millen, Jon, "Summary: Formal Models and Security", from the Workshop on Formal Verification, Menlo Park, CA, 21-23 April 1980, published in ACM Software Engineeering Notes, Vol. 5, No. 3, July 1980.

MOR80 Morse, Stephen, A Comparison of Risk Assessment Methodologies, System Development Corporation, TM-WD-7999/202/00, 30 June 1980.

NAG80 Nagle, John, Presentation on the Kernelized Secure Operating System (KSOS-11) at the Third Seminar on the DoD Computer Security Initiative, Ford Aerospace and Communications Corp., 20 November 1980.

REFERENCES

NAV79 Information Security Program Regulation, Department of the Navy, Office of the Chief of Naval Operations, OPNAV INSTRUCTION 5510.1F, 9 January 1979.

NEM78 Neumann, Peter G., "Computer System Security Evaluation", National Computer Conference Proceedings, AFIPS Press, 1978.

NEM80 Neumann, Peter G., "Summary: VERKshop Conclusions", from the Workshop on Formal Verification, Menlo Park, CA, 21-23 April 1980, published in ACM Software Engineering Notes, Vol. 5, No. 3, July 1980.

NEN78 Neumann, Frederick L., Richard J. Boland, Jeffrey Johnson, Case Studies in Computer Control and Auditing, The Touche Ross Foundation, June 1978.

NEU80 Neugent, William, AUTODIN II Security Recertification/Reaccreditation Procedures, Defense Communications Agency, September 1980.

NEU82 Neugent, William, "Acceptance Criteria for Computer Security," National Computer Conference Proceedings, AFIPS Press, 1982.

NIB79 Nibaldi, Grace H., Proposed Technical Evaluation Criteria for Trusted Computer Systems, MITRE Corporation, MITRE Report M79-225, October 25, 1979.

NIE80 Nielsen, Norman R., Brian Ruder, "Computer System Integrity Vulnerability", Information Privacy, Vol. 2, No. 1, January 1980.

NRC80 Automated Information Systems Security Program for Sensitive Data, U.S. Nuclear Regulatory Commission NRC Manual, Bulletin No. 2101-15, January 10, 1980.

OMB78 Security of Federal Automated Information Systems, Office of Management and Budget (OMB) Circular No. A-71 (Transmittal Memorandum No. 1), effective July 27, 1978.

OMB81 Internal Control Systems, OMB Circular A-123, October 28, 1981.

OMB83 Internal Control Systems, OMB Circular A-123 Revised, August 16, 1983.

OTO79 O'Toole, Thomas, Jim Schefter, "The Bumpy Road That Led Man To the Moon", The Washington Post, 15 July 1979.

OTT76 Ott, Wayne R., Thom, Gary, C., "A Critical Review of Air Pollution Index Systems in the United States and Canada", Journal of the Air Pollution Control Association, Vol. 26, No. 5, May 1976.

PAR78 Parker, Donn B., "Computer Security Differences for Accidental and Intentionally Caused Losses", National Computer Conference Proceedings, AFIPS Press, 1978.

PAR80 Parker, Donn B., "Computer and Data Abuse", Computer Security Manual, 1980.

PER76 Perry, William E., "The State-of-the-Art in EDP Auditing", EDPACS, July 1976.

PER77 Perry, William E., "Computer Audit Practices", EDPACS, July 1977.

PER78 Perry, William E., "Selecting Computer Audit Practices", EDPACS, March 1978.

PER80 Perry, William E., Internal Controls (2 Vols.), FTP Technical Library, 1980.

PER81 Perry, William E. (principal contributor), Auditing Computer Systems, Faim Technical Library, January 1980.

PMM80 Data Processing Security Evaluation Guide, Peat Marwick Mitchell & Co. (proprietary), August 1980.

POW82 Powell, Patricia B. (Editor), Software Validation, Verification, and Testing Technique and Tool Reference Guide, NBS Special Publication 500-93, September 1982.

ROE39 Roethlisberger, F. J., Management and the Worker, Cambridge, Massachusetts, 1939.

RUD78 Ruder, Brian, J. D. Madden, Robert P. Blanc (Editor), An Analysis of Computer Security Safeguards for Detecting and Preventing Intentional Computer Misuse, NBS SP 500-25, January 1978.

RUT77 Ruthberg, Zella G., Robert G. McKenzie,(Eds), Audit and Evaluation of Computer Security, Proceedings of NBS Invitational Workshop; March 22-24, 1977; NBS Special Publication 500-19, October 1977.

RUT80 Ruthberg, Zella G. (Ed), Audit and Evaluation of Computer Security II: System Vulnerabilities and Controls; Proceedings of the NBS Invitational Workshop, November 28-30, 1978; NBS Special Publication 500-57, April 1980.

SAG80 Sage, Andrew P., and Elbert B. White, "Methodologies for Risk and Hazard Assessment: A Survey and Status Report", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-10, No. 8, August 1980.

SDC79 Risk Assessment Methodology, System Development Corporation, TM-WD-7999/001/03, July 1979.

SMI80 Smith, Billy E., "Managing the Information Systems Audit: A Case Study", The Institute of Internal Auditors, Inc., 1980.

SOR79 Sorkowitz, Alfred R., "Certification Testing: A Procedure to Improve the Quality of Software Testing", Computer, August 1979.

SPE75 Spetzler, Carl S., Stael Von Holstein, Carl-Anel S., "Probability

Encoding in Decision Analysis", Management Science, Vol. 22, No. 3, November 1975.

SUL77 Sullivan, William G., W. Wayne Claycombe, Fundamentals of Forecasting, Reston Publishing Company, Inc., 1977.

TRO80 Trotter, E. T., P. S. Tasker, Industry Trusted Computer System Evaluation Process, MITRE Technical Report MTR-3931, 1 May 1980.

USA79 Automated Systems Security, US Army Regulation 380-380, 15 April 1979.

WAL79 Walker, Stephen T., "DoD Computer Security Initiative", Proceedings of the Seminar on the DoD Computer Security Initiative Program, July 17-19 1979.

WAL80 Walker, Stephen T., "The Advent of Trusted Computer Operating Systems", Program. Second Seminar on the Department of Defense Computer Security Initiative, January 15-17 1980.

WAR79 Ware, Willis [Ed], Security Controls for Computer Systems; Report of Defense Science Board Task Force on Computer Security, The Rand Corporation, Reissued October 1979.

WEB76 Webb, Doug A., W. G. Frinkel et al, "Handbook for Analyzing the Security of Operating Systems", Lawrence Livermore Laboratory (LLL), 1 November 1976.

WEI73 Weissman, Clark, "System Security Analysis/Certification Methodology and Results", System Development Corporation SP-3728, 8 October 1973.

WEI78 Weissman, Clark, The Role of Computer Technology in Fraud Protection, Presentation made at the Secretary's National Conference on Fraud, Abuse, and Error sponsored by the Department of Health, Education, and Welfare, 14 December 1978.

WEI80 Weissman, Clark, System Development Corporation, private communication.

WIL80 Wilkins, Barry J., The Internal Auditor's Information Security Handbook, The Institute of Internal Auditors, Inc., 1979.

ZAD75 Zadeh, L. A., K. S. Fu, K. Tanaka, and M. Shimara (Eds), Fuzzy Sets and Their Applications to Cognitive and Decision Processes, Academic Press, New York, 1975.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1985-461-105/20255

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBS/SP-500/133	2. Performing Organ. Report No.	3. Publication Date October 1985

4. TITLE AND SUBTITLE Computer Science and Technology:
Technology Assessment: Methods for Measuring the Level of Computer Security

5. AUTHOR(S)
William Neugent, John Gilligan, Lance Hoffman, and Zella G. Ruthberg

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)
NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
~~WASHINGTON, D.C. 20234~~
Gaithersburg, MD 20899
, System Development Corporation,
George Washington University

7. Contract/Grant No.
NB80SBCA0323

8. Type of Report & Period Covered
Final, 1980-81

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)
National Bureau of Standards
Department of Commerce
Gaithersburg, MD 20899

10. SUPPLEMENTARY NOTES
Library of Congress Catalog Card Number: 85-600600

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

This document is a companion to FIPS PUB 102, "Guideline for Computer Security Certification and Accreditation." Since a security certification depends upon a technical security evaluation, this document is meant to provide information on and insight about twenty-five evaluation methods in common use today in the security, EDP audit, and risk analysis communities. To enhance the understanding of the subject of security evaluation, the major characteristics, similarities, and differences are discussed. In addition, the significance of control groupings and exposure groupings to such analyses is brought out. The relation to environmental factors and security policy is also touched upon. The document contains a fairly extensive bibliography as well.

(The original document from the contractor has been enhanced and edited by Zella G. Ruthberg.)

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)
computer security certification; computer security evaluation; control grouping; EDP audit; environmental factors; exposure grouping; internal control; risk assessment; security control; security evaluation methodology; security policy

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☒ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES
216

15. Price

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SCIENCE & TECHNOLOGY

Superintendent of Documents,
Government Printing Office,
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name _____

Company _____

Address _____

City _____ State _____ Zip Code _____

(Notification key N-503)

NBS Technical Publications

Periodical

Journal of Research—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Bureau of Standards
Gaithersburg, MD 20899