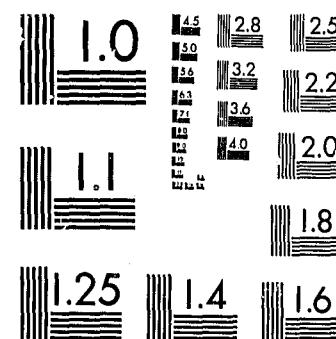


National Criminal Justice Reference Service

ncjrs

This microfiche was produced from documents received for inclusion in the NCJRS data base. Since NCJRS cannot exercise control over the physical condition of the documents submitted, the individual frame quality will vary. The resolution chart on this frame may be used to evaluate the document quality.



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Microfilming procedures used to create this fiche comply with the standards set forth in 41CFR 101-11.504.

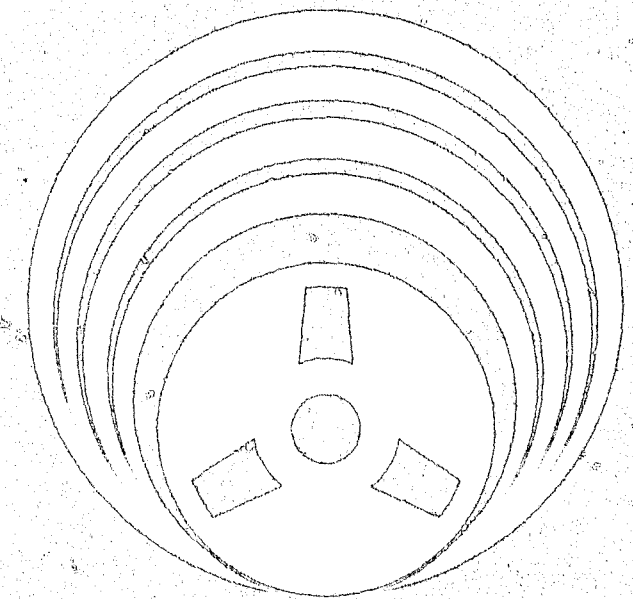
Points of view or opinions stated in this document are those of the author(s) and do not represent the official position or policies of the U. S. Department of Justice.

National Institute of Justice
United States Department of Justice
Washington, D. C. 20531

DATE FILMED

June 10, 1981

COMPUTER SCIENCE & TECHNOLOGY:



Audit and Evaluation of Computer Security II: System Vulnerabilities and Controls

76253



NBS Special Publication 500-57

**U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards**

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards¹ 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, and the Institute for Computer Sciences and Technology.

THE NATIONAL MEASUREMENT LABORATORY provides the national system of physical and chemical and materials 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; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities² — Radiation Research — Thermodynamics and Molecular Science — Analytical Chemistry — Materials Science.

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 Engineering² — Mechanical Engineering and Process Technology² — Building Technology — Fire Research — Consumer Product Technology — Field Methods.

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.

¹Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Washington, DC 20234.

²Some divisions within the center are located at Boulder, CO 80303.

COMPUTER SCIENCE & TECHNOLOGY:

Audit and Evaluation of Computer Security II: System Vulnerabilities and Controls

Proceedings of the NBS Invitational Workshop
held at Miami Beach, Florida, November 28-30, 1978

Editor:

Zella G. Ruthberg

General Co-Chairpersons:

Robert G. McKenzie

General Accounting Office
Washington, DC 20548

Session Chairpersons:

Donald L. Scantlebury
Richard D. Webb
Richard J. Guiltinan
William H. Murray
Jerry FitzGerald
Theodore M. P. Lee
Gerald E. Short
Hart Will

Zella G. Ruthberg

Institute for Computer Sciences
and Technology
National Bureau of Standards
Washington, DC 20234



U.S. DEPARTMENT OF COMMERCE, Philip M. Klutznick, Secretary

Luther H. Hodges, Jr., Deputy Secretary

Jordan J. Baruch, Assistant Secretary for Productivity, Technology, and Innovation

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued April 1980

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.

National Bureau of Standards Special Publication 500-57

Nat. Bur. Stand. (U.S.), Spec. Publ. 500-57, 210 pages (Apr. 1980)
CODEN: XNBSAV

Library of Congress Catalog Card Number: 80-600034

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1980

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
Price \$6.00

(Add 25 percent additional for other than U.S. mailing)

FOREWORD

The use of computers by Government and private organizations for the storage and manipulation of records of all kinds has continued to increase at a rapid rate in the three years since the first NBS-sponsored/GAO-supported invitational workshop on audit of computer security in March of 1977. The needs of the individual as well as Government and private organizations for the security of sensitive data and its processing - including accuracy, reliability, timeliness, and confidentiality - have therefore continued to remain a major concern to the public at large during this time interval.

In response to this need, Government laws and regulations in this arena have continued to grow and place legal requirements on computer systems. For example:

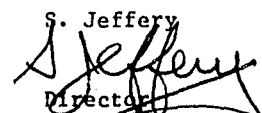
- o The Privacy Act (1974), which specified the appropriate handling of personal records by Federal agencies, has been followed by the Foreign Corrupt Practices Act (1977) that specifies the objectives of a public organization's system of internal accounting controls and, by implication, its system of general management controls.
- o The Office of Management and Budget (OMB), in its Circular A-71, Transmittal Memorandum #1 (1978), established requirements for Federal agencies to have a computer security program and appropriate audits of that security.
- o The U. S. General Accounting Office (GAO), as a direct consequence of the session on Internal Audit Standards at the first invitational workshop on audit of computer security, developed and issued in March of 1979 a set of three supplemental audit standards to help Government auditors effectively perform audits of computer-based systems.

As a consequence, the establishment of processes and procedures for controlling computer systems from the physical, administrative, and technical viewpoints has continued to expand in importance, both to the computer community in general and the Institute for Computer Sciences and Technology of the National Bureau of Standards (NBS) in particular.

In recognition of these growing needs and legal requirements for computer security and the adequate auditing of computer security, NBS, with the support of GAO, sponsored this second invitational workshop on audit of computer security in Miami Beach, Florida, on November 28-30, 1978. Following the successful approach used in the first workshop, leading experts in the audit and computer communities were again invited to share their views - this time on a more focused arrangement of the subject. Three managerial and five technical sessions on vulnerabilities and countering controls were the result. These Proceedings contain the findings of these eight sessions.

The Co-Chairpersons of this workshop were Robert G. McKenzie, an Audit Manager with the GAO [now Eastern Region Director of Audit for the National Aeronautics and Space Administration], and Zella G. Ruthberg, a Computer Scientist with NBS. The GAO again gave their generous support to this important undertaking by allowing Mr. McKenzie to devote time to the planning and execution of the workshop, by sending several vital attendees, and by providing us with Mr. Donald L. Scantlebury, Director of the Financial and General Management Studies Division, as Keynote speaker and Chairperson of the session in "Managerial and Organizational Vulnerabilities and Controls - Staff Level."

The Proceedings represent the thinking of the invited participants. The views expressed do not necessarily reflect those of the National Bureau of Standards, the U. S. General Accounting Office, or any of the organizations that sponsored an individual at the workshop. However, we at the National Bureau of Standards think these Proceedings warrant careful consideration by all those seriously concerned with security of computer systems and data.

S. Jeffery

Director
Center for Programming Science
and Technology

ABSTRACT

The National Bureau of Standards, with the support of the U.S. General Accounting Office, sponsored a second invitational workshop on computer security audit, entitled "Audit and Evaluation of Computer Security II: System Vulnerabilities and Controls," in Miami Beach, Florida, on November 28-30, 1978. A cross-section of highly qualified people in the computer science and EDP audit fields was assembled to develop material that would be directly usable for a Federal Information Processing Standard (FIPS) Guideline on the subject. In order to cover the material in a systematic fashion, the workshop was partitioned into three management sessions and five technical sessions. The management sessions addressed Managerial and Organizational Vulnerabilities and Controls at the Staff Level (1 session) and the Line Level (2 sessions). The technical sessions addressed vulnerabilities and controls in the areas of Terminal and Remote Peripherals, Communication Components, Operating Systems, Applications and Non-Integrated Data Files, and Data Base/Data Base Management Systems. These Proceedings are the reports developed by the eight sessions of the workshop.

Key Words: Applications controls, computer vulnerabilities, data base controls, data base management systems controls, EDP audit, internal audit, operating system controls, system controls, system vulnerabilities, terminal controls.

ACKNOWLEDGEMENTS

The success of the workshop can be attributed to the efforts of many people. In addition to thanking all the participants, I would particularly like to thank my Co-Chairperson, Robert G. McKenzie, for his untiring efforts in making the workshop a reality; the Session Chairpersons and Recorders for their efforts in producing the fire set of reports contained in these proceedings; Richard Canning for his excellent coordination activities while the workshop sessions were in progress; Dennis K. Branstad for the photographs taken during the workshop; and Tina Faecke for her steadfast typing support.

Z. G. Ruthberg, Editor

TABLE OF CONTENTS

FOREWORD 1fi

ABSTRACT iv

ACKNOWLEDGEMENTS iv

EXECUTIVE SUMMARY xiii

PART I: INTRODUCTION 1-1

1. HOST WELCOMING ADDRESS 1-1

2. THE CHARGE TO ALL THE SESSIONS 1-3

3. EDITOR'S COMMENTS ON THE SESSIONS AND THE REPORTS 1-10

3.1 Definitions of Terms 1-10

3.2 Materials Distributed at the Workshop 1-11

3.3 Reading the Report 1-11

3.4 References 1-11

PART II: KEYNOTE ADDRESS 2-1

1. INTRODUCTION 2-2

2. COMPUTERS HAVE BROUGHT NEED FOR NEW SECURITY CONTROLS 2-2

2.1 Old Controls are Obsolete 2-3

2.2 New Controls Are Needed 2-3

3. PURPOSE OF THIS WORKSHOP 2-3

3.1 Session 1 -- Managerial and Organizational Vulnerabilities and Controls -- Staff Level 2-4

3.2 Session 2 -- Managerial and Organizational Vulnerabilities and Controls -- Line Level - Data Processing 2-4

3.3 Session 3 -- Managerial and Organizational Vulnerabilities and Controls -- Line Level - General 2-4

3.4 Session 4 -- Terminals and Remote Peripherals 2-5

3.5 Session 5 -- Communication Components 2-5

3.6 Session 6 -- Operating Systems, and Nearby Peripherals 2-5

3.7 Session 7 -- Applications and Non-Integrated Data Files 2-6

3.8 Session 8 -- Data Base and Data Base Management Systems 2-6

4. CHALLENGE TO THE WORKSHOP 2-6

PART III: SESSION 1 -- MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS -- STAFF LEVEL 3-1

EDITOR'S NOTE 3-2

1. INTRODUCTION 3-3

1.1 General 3-3

1.2 Importance of Computer Security 3-3

1.3 Top Management Responsibilities 3-3

2. ORGANIZATIONAL STRUCTURE 3-4

2.1 General 3-4

2.2 Designate A Responsible Official 3-4

2.3 Establish an Automated Information Systems Committee 3-4

2.4 Assign Audit Responsibilities 3-4

2.5 Assign Responsibility for Personnel Security Checks 3-4

3. POLICY AND CONTROL STANDARDS	3-6
3.1 General	3-6
3.2 Assessing Security Safeguards	3-6
3.3 Establishing Control Standards	3-6
3.4 Require a Plan to Implement Controls	3-7
3.5 Establish Personnel Security Policies	3-7
4. ALLOCATE RESOURCES	3-7
4.1 General	3-7
5. REPORT ON SECURITY	3-8
5.1 General	3-8
Fig. 1 Responsibilities and Duties to Provide Security for Data Processing System ..	3-5

PART IV: SESSION 2 -- MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS --
 LINE LEVEL - DATA PROCESSING

EDITOR'S NOTE	4-2
1. INTRODUCTION	4-3
1.1 The Charge	4-3
1.2 Intent of Paper	4-3
1.3 Elements of Discussion	4-3
1.4 Comments	4-3
2. DATA PROCESSING	4-4
2.1 General	4-4
2.2 Emergency Back-up and Recovery	4-4
2.3 Security Management	4-5
2.4 Management and Controls Reports	4-5
2.4.1 Production Failure Report	4-5
2.4.2 On Time Delivery of Reports	4-5
2.4.3 Performance Monitoring Reports	4-6
2.5 Equipment Acquisition	4-6
2.6 Hardware and Software Assurance	4-6
2.7 Training	4-6
2.8 Organizational Structure and Supervision	4-7
2.9 Operating Standards and Procedures	4-7
2.9.1 Job Initiation	4-7
2.9.2 Input/Output	4-7
2.9.3 Scheduling and Job Control	4-7
2.9.4 Logs	4-7
2.9.5 Operating Systems	4-7
2.9.6 Application Programs	4-7
2.9.7 Machine Operations Standards	4-8
2.10 Personnel	4-8
2.10.1 Continuing Checks	4-8
2.10.2 Minimizing Discontent	4-8
2.11 Hardware and Software Maintenance	4-8
2.11.1 Other Important Considerations	4-8
2.12 Insurance	4-9
3. OPERATIONS	4-9
3.1 General	4-9
3.1.1 Functional Responsibilities	4-9
3.1.2 Control Requirements	4-9
3.2 Particular Policies and Procedures in Functional Areas	4-10
3.2.1 Data Entry	4-10
3.2.2 Machine Operation	4-10
3.2.3 Library Operation	4-10
3.2.4 Machine Utilization	4-11
3.2.5 Output Handling	4-11
3.2.6 Environment Control	4-11
3.2.7 Access Control	4-11
4. DATA ADMINISTRATION	4-11
4.1 General	4-11

4.2 Access Policies	4-12
4.3 Unauthorized Statistical Disclosure	4-12
4.4 Transaction Trail	4-13
4.4.1 Automated Journals	4-13
4.5 Data Checks and Integrity	4-13
5. APPLICATIONS INTERFACE	4-14
5.1 General	4-14
5.2 System Development Controls	4-14
5.3 Project Definition	4-14
5.4 System Design	4-15
5.5 Detailed Design and Programming	4-15
5.6 System Testing	4-15
5.7 Conversion	4-16
6. INTERNAL CONTROL	4-16
6.1 General	4-16
6.2 Control Policies	4-16
6.3 Control Procedures	4-17
7. HARDWARE SUPPORT	4-17
7.1 General	4-17
7.2 Introduction	4-17
7.3 Control Policies	4-18
8. BIBLIOGRAPHY	4-19

PART V: SESSION 3 -- MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS --
 LINE LEVEL - GENERAL

EDITOR'S NOTE	5-2
1. INTRODUCTION	5-3
2. APPROACH	5-3
2.1 Operational Divisions - (Section 4)	5-3
2.2 Information System Project Management - (Section 5)	5-3
2.3 Data Handling - (Section 6)	5-4
2.4 Application Program Development - (Section 7)	5-4
2.5 Data Communications - (Section 8)	5-4
2.6 Program Validation	5-4
3. CONCLUSIONS	5-4
4. OPERATIONAL DIVISIONS	5-6
4.1 Objective - Long-Range Planning	5-6
4.1.1 Nature of the Objective	5-6
4.1.2 Risks	5-6
4.1.3 Illustrative Control Procedures	5-7
4.2 Objective 2 - Short-Range Planning (Budgets)	5-7
4.2.1 Nature of the Objective	5-7
4.2.2 Risks	5-7
4.2.3 Illustrative Control Procedures	5-7
4.3 Objective 3 - System Contingency Planning	5-8
4.3.1 Nature of the Objective	5-8
4.3.2 Risks	5-8
4.3.3 Illustrative Control Procedures	5-8
4.4 Objective 4 - Organizational Communications	5-9
4.4.1 Nature of the Objective	5-9
4.4.2 Risks	5-9
4.4.3 Illustrative Control Procedures	5-9
4.5 Objective 5 - Personnel Administration	5-9
4.5.1 Nature of the Objective	5-10
4.5.2 Risks	5-10
4.5.3 Illustrative Control Procedures	5-10
5. INFORMATION SYSTEM PROJECT MANAGEMENT	5-10
5.1 Objective 1 - User Involvement in System Design Activity	5-10
5.2 Objective 2 - User Specification of Controls	5-11
5.3 Objective 3 - Continuing User Satisfaction	5-11

5.4	Objective 4 - User Compliance With External Requirements	5-11
5.5	Risks	5-11
5.6	Illustrative Control Procedures	5-11
5.6.1	System Development Life Cycle or Other Systems Development Methodology	5-12
5.6.2	User Involvement in System Development	5-12
5.6.3	User Involvement in System Changes and Maintenance	5-12
5.6.4	Design and Implementation of User Controls	5-12
6.	DATA HANDLING	5-12
6.1	Objective 1 - Maintenance of integrity of input data	5-13
6.1.1	Nature of the Objective	5-13
6.2	Objective 2 - Correct and timely reporting of exceptions	5-13
6.2.1	Nature of the Objective	5-13
6.3	Objective 3 - Secure information storage, retrieval and use	5-13
6.3.1	Nature of the Objective	5-14
6.4	Objective 4 - Controlled dissemination and storage of information	5-14
6.4.1	Nature of the Objective	5-14
6.5	Risks	5-14
6.6	Illustrative Control Procedures	5-15
7.	APPLICATION PROGRAM DEVELOPMENT	5-15
7.1	Objective 1 - Program Development Standards	5-15
7.2	Objective 2 - System Development Life Cycle Check-Points	5-15
7.3	Objective 3 - Coordination with Organizational Plans	5-16
7.4	Objective 4 - Project Management System Control	5-16
7.5	Objective 5 - Testing and Review	5-16
7.6	Risks	5-16
7.7	Illustrative Control Procedures	5-16
8.	DATA COMMUNICATIONS	5-17
8.1	Objective 1 - Integrity of Data Transmitted	5-17
8.1.1	Nature of the Objective	5-17
8.1.2	Risks	5-17
8.1.3	Illustrative Control Procedures	5-17
8.2	Objective 2 - System Security	5-17
8.2.1	Nature of the Objective	5-17
8.2.2	Risks	5-18
8.2.3	Illustrative Control Procedures	5-18
8.3	Objective 3 - System Reliability	5-18
8.3.1	Nature of the Objective	5-18
8.3.2	Risks	5-18
8.3.3	Illustrative Control Procedures	5-18
9.	SUMMARY - THE CASCADING EFFECT OF MANAGEMENT RISK	5-19
9.1	Organization Mission Impacts	5-19
9.2	Information Reliance Impacts	5-19
9.3	Control Disciplines	5-19
9.4	Organization Disciplines	5-19
9.5	Conclusions	5-19
Figure 1	Cascading Effect of Management Risk: IF - THEN	5-20,21
PART VI:	SESSION 4 -- TERMINALS AND REMOTE PERIPHERALS	6-3
EDITOR'S NOTE	6-2
1.	TASK AND ASSUMPTIONS	6-3
1.1	Task	6-3
1.2	Audience	6-3
1.3	Useful Life	6-3
1.4	Limitations	6-3
1.4.1	Remote Only	6-3
1.4.2	Terminal Selection	6-3
2.	CHARACTERISTICS OF THE REMOTE TERMINAL ENVIRONMENT	6-4
2.1	General	6-4
2.2	Application	6-4

2.3	Quantity of Terminals	6-4
2.4	Terminal Characteristics	6-4
2.4.1	Portability	6-4
2.4.2	Bandwidth	6-4
2.4.3	Storage	6-4
2.4.4	Value	6-4
2.4.5	Construction, Modularity and Assembly	6-5
2.4.6	Intelligence	6-5
2.4.7	Emanations	6-5
2.4.8	Media	6-5
3.	VULNERABILITIES	6-5
3.1	Targets	6-5
3.1.1	Vulnerabilities of Data and Programs	6-5
3.1.2	Vulnerabilities of Terminals	6-7
3.1.3	Vulnerabilities of Media	6-7
3.1.4	Vulnerability of Services	6-9
3.2	Hazards	6-10
3.2.1	Natural Hazards	6-10
3.2.2	Man-made Hazards	6-10
4.	CONTROLS	6-11
4.1	Control Principles	6-11
4.1.1	Separation of Duties	6-11
4.1.2	Restrict Access	6-12
4.1.3	Independent Authorization	6-12
4.1.4	Individual Accountability	6-12
4.1.5	Test of Concealment	6-12
4.1.6	Test of Sensitive Combinations	6-12
4.2	Control Measures	6-12
4.2.1	Explicit Assignment of Responsibility	6-12
4.2.2	Physical and Environmental Controls	6-13
4.2.3	Access Control	6-13
4.2.4	Audit Trail	6-13
4.2.5	Contingency Plans	6-14
4.2.6	Test and Reconciliation	6-15
Figure 1	Media Types	6-8
PART VII:	SESSION 5 -- COMMUNICATION COMPONENTS	7-1
EDITOR'S NOTE	7-2
Audit and Control of Communication Components	7-3
o	INTRODUCTION	7-3
o	DEFINITION OF THE COMMUNICATION COMPONENT SECURITY AUDIT	7-4
o	THE CONTROL MATRIX	7-4
o	INTERRELATIONS OF SECURITY CONTROLS	7-4
o	DEFINITION OF THE VULNERABILITIES	7-6
o	DEFINITIONS OF THE CONTROLS	7-7
o	GENERAL DEFINITIONS OF COMPONENTS	7-9
Figure I	Network Configurations	7-3
Figure II	The Control Matrix	7-5
Table 1	Exposures	7-6
PART VIII:	SESSION 6 -- PROCESSORS, OPERATING SYSTEMS AND NEARBY PERIPHERALS	8-1
EDITOR'S NOTE	8-2
o	OUTLINE	8-3
1.	PURPOSE	8-4
2.	SCOPE	8-4

3. FINDINGS	8-5
3.1 Lack of Problem Awareness	8-5
3.2 Lack of Policy	8-6
3.3 Lack of Technical Skills	8-7
3.4 Inertia Problem	8-7
4. RECOMMENDATIONS	8-9
4.1 Characterize the Problem	8-9
4.1.1 State of Current Evaluations	8-9
4.1.2 Vulnerabilities List	8-10
4.1.3 Design Principles	8-10
4.1.4 Technology Transfer	8-10
4.2 Formulate Policy	8-10
4.2.1 Aspects of DoD Policy	8-11
4.2.2 Topics to be Covered	8-12
4.3 Establish Evaluation/Accreditation Process	8-13
5. EVALUATION/ACCREDITATION PROCESS	8-14
5.1 Security Metric	8-15
5.1.1 Overview	8-15
5.1.2 Specific Features	8-17
5.1.3 Architectural Features	8-21
5.2 Evaluation Matrix	8-22
5.3 Approved Products List	8-23
5.4 Administrative Aspects	8-24
6. SPECIAL SOLUTIONS	8-25
6.1 Periods Processing	8-25
6.2 Automated Periods Processing	8-25
6.3 Secure Distributed Processing	8-25
6.4 Secure Subsystems	8-26
6.5 Assurance of Special Solutions	8-26
7. REFERENCES	8-27
Figure 1 Security Metric	8-16

PART IX: SESSION 7 -- APPLICATIONS AND NON-INTEGRATED DATA FILES 9-1

EDITOR'S NOTE 9-2

An Approach to Identification and Audit of Vulnerabilities and Controls
in Application Systems 9-3

1. INTRODUCTION	9-3
1.1 Complexity of Problem	9-3
1.2 Scope of Report	9-4
2. DEFINITIONS AND ASSUMPTIONS	9-4
2.1 Overall Workshop Assumptions	9-5
2.2 Session Assumptions	9-6
3. THE MATRIX APPROACH	9-6
3.1 Application System Resources/Assets	9-7
3.2 Application System Concerns/Exposures	9-7
3.3 Application System Controls	9-8
3.4 Limitations of the Matrix Approach	9-11
4. THE NBS APPROACH	9-11
4.1 Vulnerabilities and Security Control Objectives	9-12
4.1.1 An Example	9-12
4.1.2 Types of Vulnerabilities	9-12
4.1.3 Six Control Categories	9-13
4.2 The System Life Cycle	9-13
4.2.1 The Initiation Phase	9-13
4.2.2 The Development Phase	9-14
4.3 Limitation of this Approach	9-15
5. THE SESSION APPROACH	9-15
5.1 Arthur Andersen and Company (AA&Co.) Control Objectives Approach	9-16

5.1.1 Business Cycles as an Auditing Framework	9-16
5.1.2 Accounting System Control Objectives	9-16
5.1.3 Cycle Control Objectives	9-17
5.1.4 Advantages of the Control Objectives Approach	9-17
5.2 Transaction Flow	9-22
5.3 An Approach Towards Using Control Objectives and Transaction Flow	9-22
5.4 Additional Considerations	9-23
6. CONCLUSIONS	9-23
REFERENCES	9-24
APPENDIX A: APPLICATION SYSTEM VULNERABILITIES	9-A-1
APPENDIX B: TOWARD ESTABLISHING A SYSTEM OF CONTROLS ON SOFTWARE INTEGRITY	9-B-1
Figure 1 Program/Computer Processing Control Matrix	9-9
Figure 2 Relation of Systems Control Objectives to Transaction Flow	9-18
Figure 3 SRI Transaction Flow	9-19
Figure 4 Applications Systems Controls	9-20,21

PART X: SESSION 8 -- DATA BASE AND DATA BASE MANAGEMENT SYSTEMS 10-1

EDITOR'S NOTE 10-2

Audit, Control, and Security of Data Base and Data Base Management Systems 10-3

1. INTRODUCTION	10-3
2. DATA BASE ENVIRONMENT	10-3
2.1 Information Processing Framework	10-3
2.2 Security Audit Framework	10-3
2.2.1 Management Responsibilities	10-3
2.2.2 Current Technology Constraints	10-6
2.2.3 State of the Art Constraints	10-6
3. MULTI-LEVEL SECURITY ISSUES	10-7
3.1 Implementation Issues relevant to Secure Data Management Systems	10-7
3.1.1 The Data Base Management System as an Operating System	10-7
3.1.2 Provision for General Programming Capabilities	10-8
3.1.3 System Extensibility	10-8
3.2 Threats from Within and Threats from Without	10-8
3.3 Security and Inference Problems	10-9
3.4 Audit Trails and User Accountability	10-9
3.5 Independent Access for Auditors a Threat	10-10
3.6 Possible Data Base Management System Architecture	10-10
3.6.1 Secure Host Operating System	10-10
3.6.2 Kernelized Secure Data Management System	10-11
3.6.3 Back-End Data Management System	10-11
3.6.4 Secure Subsystem Approach	10-11
3.6.5 Encryption	10-12
3.7 Data Classification Schemes	10-12
3.7.1 Global by Data Base	10-12
3.7.2 Global by Record	10-13
3.7.3 Global by Field	10-13
3.7.4 Privileged Program Controls	10-13
3.7.5 Formulary	10-13
4. CONTROL OBJECTIVES	10-14
4.1 General Control Objectives	10-14
4.1.1 Objective #1 - Data Base Access Control	10-14
4.1.2 Objective #2 - Computer Access Control	10-14
4.1.3 Objective #3 - Software Analysis	10-14
4.1.4 Objective #4 - Security Profiles	10-14
4.1.5 Objective #5 - Data Description as Need-to-Know Control	10-15
4.1.6 Objective #6 - Data Administration	10-15
4.1.7 Objective #7 - Control Over Special DBMS Functions	10-15
4.1.8 Objective #8 - Control Over Language Use	10-15
4.1.9 Objective #9 - Validity Controls	10-16

4.1.10 Objective #10 - Data Sharing Controls	10-16
4.1.11 Objective #11 - Consistency Controls	10-16
4.1.12 Objective #12 - Recovery Controls	10-17
4.2 Application Control Objectives	10-17
4.2.1 Objective #1 - Application Standards	10-17
4.2.2 Objective #2 - Internal Audit	10-17
5. RECOMMENDATIONS	10-17
5.1 The Role of the National Bureau of Standards	10-17
5.2 Independent Access Paths?	10-18
5.3 System Maintenance	10-18
REFERENCES	10-19
Figure 1 Data Base Environment	10-4
Figure 2 Security Audit Framework	10-5
APPENDIX A: GLOSSARY	A-1
APPENDIX B: WORKSHOP ATTENDEE LIST	B-1

EXECUTIVE SUMMARY

On November 28-30, 1978 the National Bureau of Standards (NBS), with the support of the U. S. General Accounting Office (GAO), held a second invitational workshop on the subject of audit for computer security in Miami Beach, Florida. The first workshop, held on March 22-24, 1977 [see NBS Special Publication 500-19, "Audit and Evaluation of Computer Security"] was an exploratory effort for determining the state-of-the-art and future areas for research, and consequently was structured into ten overlapping areas of concern. This second workshop had the more difficult goal of providing direct inputs for a Federal Information Processing Standards (FIPS) guideline on the subject and was consequently more focused in its structure.

The Co-Chairpersons selected the session topics from two broad categories that together would cover the subject systematically:

1. an organization's management concerns generally appropriate for any computer system it uses (three sessions), and
2. an organization's technical concerns appropriate for its computer systems having specific technical features (five sessions).

This yielded a total of eight sessions for the workshop. Further, since security of a computer system can be viewed as a three dimensional problem, with its operating environment, its vulnerabilities, and its countering controls as the three variables, the charge given to the eight sessions asked each group to elaborate on the vulnerabilities and countering controls suitable for a worst case environment -- that of a multi-user teleprocessing system. The worst case environment was selected in order to maximize the coverage of the vulnerabilities and controls described. (It is left for future activities in this subject area to define useful environment categories.) In addition, each session was asked to identify system vulnerabilities without regard to the risk of exploitation since risk analysis is the subject of other on-going development outside the scope of this workshop. Finally, if time permitted, the qualitative effectiveness of the controls and the cost of implementation were to be addressed. The precise charge given to the eight sessions can be found in Part I, Section 2.

By using their knowledge of people in the field and the recommendations of numerous people contacted prior to the second workshop, the Co-Chairpersons were able again to invite an outstanding group of attendees from both the audit and computer science communities. The three days of the workshop allowed each session to develop its material to a level sufficient to report its findings in outline form to the group as a whole on the last day. Each group then developed its position paper on its topic over the next several months. It is these papers that are contained in this publication.

Although the sessions worked independently of one another, except for conversations at refreshment breaks, it is the opinion of the Co-Chairpersons that the workshop is of such a structure that the reader will derive the most benefit from reading the management sections first, due to their general applicability, and the technical sections second, due to their applicability to specific components of the technical environment. Since the management sessions have addressed such organizational units as system control, application interface, data base administration, data handling, application program development, and communications from the management point of view, a second reading of this report would benefit from cross-referencing the related technical sections. The reader should also note that, due to growing awareness in this field of the need for a set of common definitions of frequently used terms, the Co-Chairpersons, with the help of persons from NBS, GAO and the attendees, have come up with a small glossary which can be found in Appendix A.

MANAGEMENT SESSIONS

The management sessions were asked to report on the managerial and organizational vulnerabilities and controls of an organization/agency computer system. The Co-Chairpersons developed an organization/agency model suitable for categorizing and grouping system vulnerabilities and management controls (See Part I, Sec. 2, p. 1-5). Fig. 2 in NBS Publication 500-25, "An Analysis of Security Safeguards for Detecting and Preventing Intentional Computer Misuse," was used as the departure point for developing this model. The organizational units in this model were then grouped into three logical sets that could each be handled comfortably by a single session at the workshop.

Session 1 was asked to consider the staff level organizational units for Internal Audit, Procurement, Personnel, and Security Administration. Session 2 was asked to address the single line level unit for Data Processing since that had many components and covered the areas of Operations, System Control, and Data Base Administration. Under System Control were placed three subunits concerning Application Interface, Internal Control, and Hardware Support. Session 3 was asked to address the line level units for Operational Divisions, for Application Program Development, and for Communications. The important areas for consideration in the Operational Divisions were specified to be Information System Project Management and Data Handling; only one area, Program Validation, was explicitly included under Application Program Development, although that unit has many more facets.

SESSION 1 (PART III): MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS - STAFF LEVEL

This group was originally asked to address the managerial and organizational vulnerabilities and controls at the staff level and to assume there exist staff level units for Internal Audit, Procurement, Personnel, and Security Administration. [See PART I, Section 2 for the complete charge given this group.] The subject was instead broadened by the group to address the responsibilities and duties of Top Management and its relation to its supporting staff and line level functions. The results of these deliberations are thus able to provide a framework within which all of the managerial and technical concerns of the other sessions can be viewed. The organizational unit for Procurement was not covered due to lack of time.

The group asserted at the outset that it is the responsibility of Top Management to establish the physical, administrative, and technical safeguards for its automated data processing systems. A chart was drawn up by the group (see Part III, Fig. 1) which shows Top Management's responsibilities and the assignment of duties recommended. The specific duties of Top Management that could not be delegated to staff or line level management are:

- 1) to provide for an organizational structure to assess vulnerabilities of, and to provide effective controls over, its data processing systems,
- 2) to establish policy and control standards which promote secure, well-controlled systems,
- 3) to allocate adequate resources to provide controls and periodically test them, and
- 4) to require periodic reports on security.

One of the salient recommendations is the designation of a Principal Assistant to Top Management who would be responsible for data processing systems and their security. Such an office would establish procedures for implementing security policies and control standards, assign responsibilities for security (including assessment of risks and safeguards), prepare contingency plans, and report on security to Top Management. The group considers the nineteen standards found in "The Auditor's Study and Evaluation of Internal Control in EDP Systems," published by the AICPA, as appropriate for implementation. The

paper lists all nineteen. It is also suggested that the Principal Assistant chair a high level organization committee to monitor, evaluate the adequacy of, and make policy recommendations concerning the organization's security controls.

The personnel security policies for screening of individuals who handle automated information systems are the responsibility of Top Management but the implementation can be assigned to Personnel. Top Management should also require its Internal Auditors to periodically assess the adequacy of controls and security safeguards for existing systems and to evaluate proposed systems at critical stages in development.

SESSION 2 (PART IV): MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS - LINE LEVEL - DATA PROCESSING

This group addressed the question of managerial and organizational vulnerabilities and countering controls for the line level unit for Data Processing. [See PART I, Section 2 for the complete charge given to this group.] The functional areas of Operations, System Control, and Data Administration were assumed to exist within this unit and the System Control area was discussed under the three functional subunits: Application Interface, Internal Control, and Hardware Support. The group decided to discuss control policies in relation to these functional areas and omit cost and effectiveness since these last two items are functions of the particular installation and environment.

There are a number of overall control policies and procedures that should be in effect for security and control purposes in every data center. The most important ones, enumerated in the paper and discussed at some length, were grouped under the following headings: emergency back-up and recovery; security management; management and control reports; equipment acquisition; hardware and software assurance; training; organizational structure and supervision; operating standards and procedures; machine operations standards; personnel security; hardware and software maintenance; and insurance.

The paper defines Operations as responsible for seven functions and discusses control policies in each of these areas. The areas are: data entry for manually received data, machine operation, library operation, machine utilization, output handling, environmental control, and access control. The policies and procedures that need to be in place across all of these functions are standard written procedures; effective supervision; preparation and review of activity logs; formal acceptance procedures for new software and hardware and for modification thereto; personnel recruitment, training, job descriptions, security clearance, privileges, and evaluation; and preparation and review of exception reports of control failures.

Data administration is viewed as responsible for successful management and controls of data files and data bases necessary to support the information processing system. It is supported by data management systems and data base management systems. The control policies revolve around the three concepts of identification, authorization, and authentication and include consideration of access policies, detection of unauthorized statistical disclosure, the maintaining of appropriate transaction trails, and the integrity of data and programs.

The rest of the paper deals with the System Control organizational unit. The Applications Interface component of System Control deals with the specification of suitable application systems programming, testing, and documentation criteria. The systems development cycle consists of the following phases, all of which require the defining of control objectives: project definition, system design, detailed design and programming, system testing, and conversion. System development controls should

- 1) be more detailed for more sensitive applications,
- 2) have a modular approach to structure and acceptance,
- 3) concurrently develop documentation,
- 4) be matched to the sensitivity of the software,

- 5) protect documentation of sensitive software, and
- 6) evaluate risks associated with all identified vulnerabilities.

The Internal Control component of System Control is responsible for cataloguing all internal controls, maintaining application system controls, and establishing and maintaining system software control policies. The control policies mentioned in the paper are: to safeguard all documentation supporting applications and systems programs, to document all system software modifications, to catalog all control security features of the operating system, to train operational personnel in the functions of both applications and systems before they become productional, to control utilities in the same manner as applications, to verify version/level controls of all production programs, to define reports for utilizing logs of unauthorized access, and to test systems for acceptance. Control procedures for implementing the above controls must also be in place and documented.

The Hardware Support component of System Control is responsible for hardware, planning, acquisition, and maintenance. The hardware support under consideration should include: central site, communications, remote processing, and off-line hardware; maintenance personnel; and administrative procedures. Some of the control policies to be considered are: optimal configuration management, optimal maintenance schedules, system incident records, personnel requirements, records of hardware changes, proper communication with personnel, and system monitoring.

SESSION 3 (PART V): MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS - LINE LEVEL - GENERAL

This group was asked to address all line level organizational units other than Data Processing. [See PART I, Section 2 for the complete charge given to this group.]

That included consideration of (1) Operational Divisions (with Information System Project Management and Data Handling as its subunits of interest here); (2) Application Program Development; and (3) Data Communications. It should be noted that the panel came up with additional subunits of its own choosing. The overall approach taken by this group was to specify, for each organizational unit or subunit, the objectives for a system of controls and the risks associated with failure to achieve them. Since specific vulnerabilities and countering controls are very dependent on a number of internal and external factors, the group elected to suggest some illustrative control procedures that might be used by management to achieve its objectives and then to reference a half dozen publications on control procedures.

Four major areas for systems management concern were identified and placed in order of importance as follows:

- 1) Organization and mission impacts
- 2) Information reliance impacts
- 3) Control disciplines
- 4) Organization disciplines.

The risks cited under the various organizational units and subunits were then grouped under these, thus creating four principal risk levels or tiers in descending order of importance. Closer study by the group showed that these risks were interrelated in a cascading fashion both upwards and downwards in these areas of concerns, i.e., a particular risk, for a particular area of responsibility (such as Information Systems Project Management or Data Communication) could be caused by a risk above it in the tiers or could cause a risk in a lower tier.

This analysis was then represented in a diagram (Part V, Fig. 1) at the end of the paper. The figure allowed for several major observations:

- 1) Data Handling is a major area for concern in Tier 1, Organizational and Mission Impacts;

- 2) Preventive actions are more available to planning activities in Operational Divisions and to Information Systems Project Management than in other areas;
- 3) Organizational Communications within Operational Divisions is the second most vulnerable managerial responsibility;
- 4) Long range business and systems plans are necessary to successfully support the shorter range budget process;
- 5) Failure of Organizational Disciplines undermines the application of Control Disciplines.

The group concluded that there were four overriding control objectives that must receive a higher level of attention than has been the case in the past. These are:

- 1) Management has the ultimate responsibility for system controls.
- 2) Users have a non-negotiable responsibility for the controls in their systems.
- 3) Short and long-term planning and budgeting within a properly designed organization structure is a key internal control.
- 4) An appropriate systems development methodology is essential to managing and maintaining the structure of control and to auditability.

TECHNICAL SESSIONS

The technical sessions were also asked to address, in their topic area, the vulnerabilities and controls of a worst-case environment - that of a multi-user teleprocessing system. An illustration of such an environment was included in the charge to these sessions (see Part I, Sec. 2, Fig 1). The groupings of components selected for consideration by the five technical sessions are: Terminals and Remote Peripherals; Communication Components; Processors, Operating Systems, and Nearby Peripherals; Applications and Non-Integrated Data Files; and Data Base and Data Base Management Systems. Each session was asked to identify the vulnerabilities in its topic area, and the controls which will deter the possible exploitation of each and/or permit detection of an actual or attempted exploitation.

SESSION 4 (PART VI): TERMINALS AND REMOTE PERIPHERALS

This session was asked to address the vulnerabilities and countering controls appropriate for remote processing without regard to risk of exploitation. The reader is initially told that consideration of communications and locally resident applications is left for the other sessions; and that appropriateness of the terminal for the security of the application is assumed. Finally, vulnerabilities and controls are viewed in the most general terms and tied to specific devices or media for illustrative purposes only. The paper is directed at auditors although managers and system designers will also find it useful.

The group identified three essential properties of the remote environment that affect the choice of controls: the application, the number of terminals, and the terminal characteristics. Not only are the control requirements highly dependent on the nature of the application, but, the more flexibility offered the end user for mixing application types, the more rigorous the needed controls. The sensitivity of the computer system, in general, also increases with increasing number of terminals, so that a multiterminal site requires more rigorous controls. The terminal characteristics that affect the sensitivity and, therefore, the choice of controls are: portability; bandwidth or character rate; amount of local storage; value; construction, modularity, and assembly; intelligence; emanations; and number and types of media supported.

Four targets within a remote terminal environment were identified as vulnerable:

- 1) data (including programs),
- 2) the terminal or device,
- 3) media (as distinct from the data recorded on it), and
- 4) the service or capacity of the system.

The vulnerabilities of data were discussed at length in relation to four characteristics of data: its location in the system, its form, its sensitivity (which depends on quantity or size, context, interpretation, and age), and its type (application, system, or program). With regard to terminals, the group concluded that they were vulnerable to damage, theft, and unauthorized use and the extent of the vulnerability depends on their characteristics. Media were seen as having the same vulnerabilities as terminals but having greater susceptibility than terminals. A variety of media types (see Part VI, Fig 1) were identified and a relevant set of media characteristics discussed. These included density, portability, size, permanence, value, integrity, authenticity, flammability, and fragility. Finally, conversion of service or capacity from the use of the owners was seen as a serious vulnerability at remote sites.

The group decided it would be useful to identify the hazards at a remote site before continuing on to a discussion of controls. Hazards could be natural or man-made. Man-made hazards could be accidental (due to errors or omissions) or intentional. Intentional man-made hazards could be viewed by type (i.e., vandalism, riots, theft, etc.) and by method of attack (i.e., browsing, exhaustive attack, Trojan horse, etc.).

In identifying controls effective against the identified vulnerabilities and hazards, the group first articulated six control principles:

- 1) separation of duties,
- 2) restriction of access,
- 3) independent authorization,
- 4) individual accountability,
- 5) test of concealment, and
- 6) test of sensitive combinations.

Based on these principles, the group elaborated on six categories of control measures, as follows:

- 1) explicit assignment of responsibility,
- 2) physical and environmental controls,
- 3) access controls,
- 4) audit trails,
- 5) contingency plans, and
- 6) test and reconciliation.

It should be noted that the section on controls is particularly addressed to the auditor.

SESSION 5 (PART VII): VULNERABILITIES OF AND CONTROLS FOR COMMUNICATIONS COMPONENTS

This group presented a set of guidelines for auditors or security experts to enable them to review the adequacy of administrative and technical controls in place in a multi-user teleprocessing environment. In order to better understand what is meant by a teleprocessing environment, a figure (see Part VII, Fig 1) was developed to show examples of alternative teleprocessing configurations. The paper contains definitions of components/characteristics found in such configurations.

A Control Matrix (see Part VII, Fig 2) was then drawn up which relates the various vulnerabilities (threats) to the specific controls to mitigate them. The Matrix contains nine vulnerabilities (threats) across the top and twenty-six controls down the left side. Within the cells of the Matrix, an X indicates the control is a primary one and an O that it is secondary in protecting against the vulnerability (threat). These vulnerabilities and controls are all defined within the paper.

The Matrix has two other uses. The first is to determine the exposures resulting whenever a vulnerability (threat) actually occurs. Nine exposures are found in Table I (lettered A-I) and the pertinent exposures listed by letter at the bottom of the Matrix for each of the nine vulnerabilities (threats). The second other use of the Matrix is to identify effective components in the network for locating the controls. These components are listed by number down the right side of the Matrix.

SESSION 6 (PART VIII): PROCESSORS, OPERATING SYSTEMS, AND NEARBY PERIPHERALS

Since the internal controls needed to produce secure operating systems today are still in the development stage, this group decided to address the following broader question:

"What authoritative ways exist, or should exist, to decide whether a particular computer system is 'secure enough' for a particular intended environment of operation, and, if a given system is not 'secure enough' for an intended application, what measures could or should be taken to make it so?"

In the course of responding to the above question, the group not only discusses the status of processor/operating system/nearby peripherals security, but introduces a unique framework within which to assess and certify the security of a computer system in general. In the course of the discussion, the major emphasis is on operating systems, but other forms of software critical to security are included.

The group concluded that computer security requires balanced attention to three subjects:

- 1) management doctrine and formal policy for security,
- 2) protection mechanisms for software and hardware,
- 3) assurance of the proper design and implementation of the protection mechanisms.

In assessing the state of affairs in the computer field and Federal government today, however, they drew the following conclusions:

- 1) There is a surprising lack of awareness that there is a "technical computer security" problem. It is commonly accepted among the knowledgeable that penetration of a system is not that difficult for a skilled individual, and that no operating system has sufficient internal controls to effectively isolate a determined user from data he is not entitled to.
- 2) Outside of the Department of Defense (DoD) and the Intelligence Community there is no well-thought-out policy about information security in the Federal government. There do not exist standard ways of categorizing and identifying sensitive information, rules and procedures for deciding who is allowed to have what kind of access to what kinds of sensitive information, or rules on practices for handling sensitive information.
- 3) Technical skills needed to analyze and provide solutions to the security problems of a given system in a given environment are not widespread. As a consequence people in the Federal government with procurement responsibility for computer systems do not generally have the skills and experience to write the computer security portions of procurement specifications.
- 4) There is an inherent inertia in the system development and procurement cycles.

Users of computers, generally speaking, do not care very much about security and vendors wait for demand before engaging in new developments. This then leads to a real slow-down in the transfer of technology.

This group came up with the recommendation that, to remedy the situation, three major tasks be pursued by a group or groups of technical and policy individuals chartered by NBS, possibly in concert with GAO or other agencies. The tasks with their subtasks are as follows:

- 1) From available literature and people's experience prepare a series of reports on the current state-of-the-art.
 - a) A candid report on all past efforts to penetrate and repair operating systems,
 - b) A report on the kinds of vulnerabilities found in current operating systems,
 - c) A report on design principles for security in operating systems,
 - d) a technology transfer report on all current research on secure operating systems.
- 2) Have OMB or GSA, with the technical recommendations and guidance of NBS, form a group to formulate security policy, practices, and doctrine for those parts of the Federal government that don't already have them. Current practices in DoD and the Intelligence Community should be used as a model. In particular, use of the following two aspects of DoD computer security policy should be considered:
 - a) An access control policy that distinguishes between mandatory and discretionary access.
 - b) A mandatory access control policy that is general enough to support both hierarchical security levels and lattice-structured sets of security categories. Further, rules for declaring data to be sensitive and for handling such data within an agency or between agencies should be formulated.
- 3) Formulate and institutionalize a process for evaluating the security of computer systems, and for accrediting particular systems for particular applications. Two preliminary steps that should be taken first are:
 - a) Develop a standard set of procurement specifications for computer security.
 - b) Develop an "approved products list" by doing a preliminary evaluation of the more popular or security-critical current systems using the security metric discussed below.

A lengthy discussion of a proposed approach for formalizing the evaluation of security of a computer system is included in the paper. Such an evaluation would be based on evaluating one extrinsic set of attributes of the system - policy - and two intrinsic sets of attributes - protection mechanisms and assurance measures. The paper only discusses the intrinsic attributes (which are technical) and leaves the extrinsic attribute for management consideration. The suggested security metric consists of an inverted bull's eye type figure that assigns numerical levels to various possible protection mechanisms and assurance measures. As one moves out from the center of the figure one finds features that afford greater and greater protection levels. With an agreed upon "bull's eye" figure, one could then determine, in a meaningful way, how "good" a system needs to be for a particular threat environment, application environment, and sensitivity of data. An audit for security would then become a test for compliance.

For the operators of current systems preparing long-range plans for security improvements of existing products or installations the group recommended and discussed the following five interim approaches for serious consideration:

- 1) Periods Processing - a method for exclusively processing one type of sensitive information on a computer for a given time period.
- 2) Automated Periods Processing - automating such exclusive processing with an auxiliary security trusted mini-computer.
- 3) Secure Distributed Processing - using a network of computers, many of which process different levels or kind of sensitive data.
- 4) Secure Subsystems - using a secure (i.e., trusted) subsystem with an untrusted operating system which is constrained by external, physical access controls.
- 5) Assurance of Special Software - using some kind of evaluation process for determining the degree of "trusted" software in 2)-4) above.

SESSION 7 (PART IX): APPLICATIONS AND NON-INTEGRATED DATA FILES

This session worked on identifying vulnerabilities and countering controls which would deter and/or detect exploitation of vulnerabilities associated with applications, application program development, application program maintenance, and non-Data Base Management System (DBMS) data files. The consensus generally agreed upon involved a transaction-flow/control-objectives approach which was a synthesis of the Arthur Andersen & Co. approach to evaluating internal controls for accounting systems and the transaction flow analysis found in the SRI Systems Auditability and Control Study. In order to augment these conclusions, the author reviews two other comprehensive approaches in the literature before discussing the session's approach. Some of the advantages and disadvantages of each are included.

The paper begins by defining several pertinent terms and then stating the session's assumptions concerning the existence of certain policies and guidelines, the limitations of controls, and the ignoring of data sensitivity. With this as a base, the first strategy, the matrix approach, is reviewed. This approach develops a detailed list of 91 controls which will protect specific resources/assets of an application from its vulnerabilities (concerns/exposures). This approach is taken from "Internal Controls for Computerized Systems" by J. Fitzgerald, and in particular from the Program/Computer Processing matrix developed by him. The book contains eight other matrices, each representing a component of the data processing function that might be reviewed.

Although the matrix approach gives the user a quick general checklist, it is still up to the user to narrowly define these variables for the application under consideration. Also, since the application system security is intimately related to its environment's security, it would be necessary to use all nine matrices (with 650 controls to check) to get a well-rounded view. Issues of control redundancy and sharing are not addressed, nor that of how to generate the interpretation of the overall security based on these "yes/no" answers.

The second strategy reviewed is one developed by NBS in their draft guideline "Security for Computer Applications." The concepts stressed here are system control objectives and partitioning the problem into phases of the application system life cycle. Undesirable computer events are classified in terms of their general effects on computerized data rather than their ultimate effects on the organization.

Three classifications of undesirable events (vulnerabilities that are activated) are related to three general security control objectives for all application systems.

These are:

<u>Vulnerability</u>	<u>Security Control Objective</u>
1. Modification or destruction of data.....	Data Integrity
2. Disclosure of data.....	Data Confidentiality
3. Unavailability of data or system service.....	ADP Availability

The NBS work does not couple vulnerabilities with specific controls. However, it does contain a lengthy list of application environment vulnerabilities in eight categories (completely listed in Appendix A of this paper) and detailed discussion of controls in six basic control categories with an indication of the general problem each will address.

Having established this framework, the NBS approach goes on to discuss the placement and use of appropriate controls at each stage of the system life cycle. The stages elaborated upon are: the initiation phase, where system requirements, objectives, and sensitivity are defined; the development phase, where activities of security requirements definition, design, programming, and testing occur; and operations phase. The author

reviews only initiation and development since those fulfilled the workshop charge.

The NBS strategy provides insight to initial application design and is useful for making major system modifications, but is not a road map for performing speedy system reviews or audits.

The session participants used the experience in a traditional audit discipline (the Arthur Andersen & Co. [AA&Co.] Guide for Studying and Evaluating Internal Accounting Controls) as their starting point. The AA&Co. approach stresses two concepts - the setting up of system control objectives and the applying of these at each step of the system's transaction flow. The paper describes the AA&Co. Guide's use of these concepts for accounting systems in general and a manufacturing company in particular.

The accounting system is viewed as divided into groupings of events - financial planning/control and a limited number of business "cycles" (i.e., treasury, expenditure, revenue, etc.). This establishes a framework for review of application programs that has manageable size pieces. The general control objectives developed are based on accepted practice and legislative requirements. The more specific control objectives for each grouping of events are derived from these. The AA&Co. Guide goes on to apply these control objectives to the transaction flow of the accounting application.

The session participants concluded that a general application can be viewed in a similar fashion. A more general transaction flow description can be found in the SRI Systems Auditability and control study where controls are grouped under six components of the flow. Figure 4 in the paper (PART IX) contains a master chart, developed by GAO, of the controls found in the SRI document, grouped in the above six categories. This master chart could be used by auditors of an application system as well as by designers (with some kind of risk assessment before implementation).

The three approaches discussed by this paper each have merit but none is complete. The recurrent theme is problem simplification and is embodied in three devices: management's early definition of overall system control objectives, partitioning the problem by life cycle considerations and transaction flow vulnerabilities, and employment of schematics such as matrices and flowcharts. It is recommended that NBS more fully develop the above approaches and further the dialogue on secure application design.

SESSION 8 (PART X): DATA BASE AND DATA BASE MANAGEMENT SYSTEMS

Since an understanding of the data base environment is essential to achieving the security of a data base or a data base management system, this paper begins by identifying and illustrating with two figures: an information processing framework and a security audit framework. The information processing framework figure (Fig. 1, PART X) shows the relation of user, languages and application programs, and the data bank (i.e., data base management system/schema/subschema and the data base), and stresses the importance of the user/language system interface and the language system/data bank interface. The security audit framework figure (Fig. 2, Part X) shows the relation of the security issues for all the components that affect the data base and the data base management system. This paper's component approach to computer security auditing assumes:

- 1) Management is responsible for the establishment and evaluation of the system controls.
- 2) A computer security audit must address the current technology being used and be based on a total system evaluation plan.
- 3) The state-of-the-art is such that security is not yet a mandatory feature of hardware, firmware, and software available in the marketplace; risk assessment is one of the main tools available for determining where the weaknesses are; and defining the sensitivity levels of the organization's data is essential for determining when these weaknesses should be corrected.

The paper then goes on to discuss the security issues of a data base management system that is multi-level, i.e., has users with different levels of clearance at the same time. The issues of fraud within a single-level group of users, and of after-the-fact analysis of fraud via audit trails is considered beyond the scope of this particular discussion.

The first set of issues discussed are those of implementation of a secure DBMS. The paper indicates reasons for caution and directions for further research, rather than presenting remedies. These issues, briefly, are:

- 1) DBMS as an Operating System - Since a DBMS is usually designed to interface with an operating system or to run on bare hardware while containing within it the functionality of an operating system, the security problems and solutions for a DBMS parallel those for operating systems. It has been shown that no general technique can be developed to prove that a system is secure; however, specific systems may be designed to be secure against known attacks. A promising approach is one in which improved security will be obtained by running under a secure (kernelized) operating system.
- 2) Provision for General Programming Capabilities - The interface of a DBMS with new application programs offers an opportunity for system penetration by users who can write and compile their own programs. Therefore, compilers and new programs should be strictly audited and controlled.
- 3) System Extensibility - For the same reason new code added by system programmers to the existing DBMS should also be strictly audited and controlled.

The next set of security issues involve users, inference, audit trails, and auditors. The points stressed are:

- 1) Users (authorized or unauthorized) can compromise a system.
- 2) Inference from combining the results of a number of queries can compromise a system.
- 3) Audit trails, if properly designed and analyzed can expose and discourage computer misuse.
- 4) Auditors cannot be permitted to have an independent access route to the system that bypasses the security enforcement mechanism.

The next broad security issue is that of possible DBMS architectures. The discussion is limited to those in which there is: a secure host operating system; a stand-alone data management system with its own security kernel; a security mediation on one computer and a set of stand-alone computers accessible from the first; a "secure" subsystem on a computer with a standard operating system; and a standard dms with a standard operating system in which the protection is provided through the use of encryption. The architectures whose salient features are discussed are: Secure Host Operating System, Kernelized Secure Data Management System, Back-End Data Management System, Secure Subsystem Approach, and Encryption.

The last broad security issue is that of data classification schemes in a multi-level data base. Comments are made on the impact on implementation mechanisms. The classification schemes discussed are: Global by Data Base, Global by Record, Global by Field, Privileged Program Controls, and Formulary.

The paper then goes on to identify and briefly discuss a number of control objectives. There are twelve general objectives and two application objectives as follows: Data Base Access Control for Users, Computer Access Control for Users, Software Analysis for Unwanted Code, Security Profiles of Users, Data Description as Need-to-Know Control (DDL), Data Administration Functions Defined, Control Over Special DBMS Functions (DBA, Auditor, etc.), Control Over Language Use (Interfaces), Validity Controls on Data, Data Having Controls (Deadlock, Lost Updates), Consistency Controls on Data, Recovery Controls (i.e., Journaling), Application Standards Enforced, Internal Audit - at irregular intervals.

Since data bases and DBMSs influence the information processing of an organization profoundly, the security, control and audit implications for the information processing cannot be separated from the data management support provided by the DBMS and operating systems software. The paper therefore recommends:

- 1) NBS should participate in the development and application of criteria for evaluating the "security trustworthiness" of DBMS, develop standards or be a catalyst for them.
- 2) Under current software design technology, there should be no independent access paths to the data for special groups such as auditors. More research is needed in this area.
- 3) There should be further studies of system maintenance in a secure environment.





Host: M. Zane Thornton



Co-Chairpersons: Zella G. Ruthberg,
Robert G. McKenzie



PART I: INTRODUCTION

1. HOST WELCOMING ADDRESS

M. Zane Thornton
Acting Director
Institute for Computer Sciences and Technology
National Bureau of Standards

I am pleased to welcome you to the second National Bureau of Standards' Invitational Workshop on Audit and Evaluation of Computer Security. I also want to thank you for your response to our call for help. This workshop, as you know, is a follow-on to the first one held at this same location in March of 1977.

The first workshop, which called upon very highly qualified individuals in the audit and computer science communities, produced a Proceedings, NBS Special Publication 500-19, which delineated the state-of-the-art and pointed to future areas for computer audit research. This document has been very well received by the EDP audit community and, from all the feed-back we have received, is being actively used in the field.

However, as stated at the first workshop, that effort was being made to develop the foundations for a set of Federal guidelines on Audit and Evaluation of Computer Security. The workshop this week, with its eight carefully chosen session topics, has been organized so that the results of its sessions directly form the basic input for a set of Federal guidelines on the subject. The attendees here today are again an impressively qualified group of professionals from the audit and computer science communities. In fact, about half of you attended the first workshop. In view of the broad range of expertise among those here today, I anticipate that your efforts this week will enjoy a success equal to or greater than that of the first workshop.

You may be interested to know that a dozen Federal agencies are represented by 40% of the attendees today. These agencies include the General Accounting Office, the Department of Health, Education, and Welfare, the General Services Administration, the Department of Defense, the Department of the Interior, the National Science Foundation, and our own Department of Commerce.

These agencies are represented here today by an impressive group of people. I'd like to call your attention particularly to the return appearances of Frank S. Sato, the Deputy Assistant Secretary of Defense for Audit, Donald L. Scantlebury, the Director of the Financial and General Management Studies Division of the General Accounting Office, and Howard R. Davia, the Director of the Office of Audit at the General Services Administration. In addition, we are fortunate to have as newcomers Joseph A. Sickon, the Director of the Office of Audits at the Department of Commerce, and Bryan B. Mitchell, an Assistant Inspector General at the Department of Health, Education and Welfare. Their collective experience will provide the staff level management session with invaluable inputs.

The remaining 60% of the attendees again come from a broad variety of accounting firms, software and hardware organizations, private industry, and universities. This time we have representation from five accounting firms, eight software organizations, two main-frame manufacturers, five banks, three insurance firms, two non-profit research organizations, five universities, a publishing house, and several major industrial corporations.

The array of expertise is similar in breadth and depth to that in the first workshop.

As you know, the subject of this workshop is an interdisciplinary one, and if one takes a second look at the attendee roster, one sees that the audit field is covered by members of the American Institute of Certified Public Accountants, the Institute of Internal Auditors, the EDP Auditors' Association, the Association of Government Accountants, the Canadian Institute of Chartered Accountants, five large accounting firms in the private sector, and auditors from various Government and private organizations. The computer aspect of this workshop is again covered by persons actively engaged in the research, development, and use of control software and techniques in industry, Government, and universities.

This run-down of the make-up of this group gives you some idea of the effort made by the Co-Chairpersons of this workshop to produce for a second time a roster of attendees able to tackle this interdisciplinary subject with vigor, originality, and productiveness. I'd like to thank Mr. Robert G. McKenzie of the General Accounting Office and Mrs. Zella G. Ruthberg of my own Institute for Computer Sciences and Technology for developing a set of sessions for this workshop that form a coherent approach to the guidelines, and then proceeding to successfully convince all of you to participate in this unique effort. Mr. McKenzie's lengthy experience in the field of computers and internal audit and Mrs. Ruthberg's computer science and organizational skills have provided the workshop with a balance of capabilities that have brought us to the beginning point of a second multifaceted collaboration on this most important topic.

As stated at the first workshop, the interest of the Institute for Computer Sciences and Technology of the National Bureau of Standards is to provide Federal agencies with standards and guidelines for information processing. Using the information gathered by the first workshop as a jumping off point, it is our hope that this workshop will enable us to produce the first set of Federal guidelines on Audit and Evaluation of Computer Security. For this purpose, we define computer security audit as an independent evaluation of the controls employed to ensure (1) the accuracy and reliability of data maintained or generated by an ADP system, (2) the appropriate protection of an organization's information assets from all significant anticipated threats or hazards, and (3) the operational reliability and performance assurance of all components of ADP systems.

Again, I'd like to thank you for taking your valuable time to assist us in this nationally needed effort. In view of the outstanding array of abilities in this room, I know that your deliberations will result in an outstanding document.

2. THE CHARGE TO ALL THE SESSIONS

The following eight pages contain the tasking instructions given to the attendees of the eight sessions of the workshop prior to the workshop. They are included here so that the reader may better understand these Proceedings which contain the session responses to the questions posed by this tasking document.

NBS INVITATIONAL WORKSHOP ON AUDIT AND EVALUATION OF COMPUTER SECURITY II: SYSTEM VULNERABILITIES AND CONTROLS

TASKING OF THE SESSIONS

INTRODUCTION

Background and Objective

This workshop is a follow-on to the first NBS Invitational Workshop on Audit and Evaluation of Computer Security which consolidated the state-of-the-art information available in the field and defined areas for future research. Again, the foremost experts in the auditing and computer science communities are being asked to address some of the most pressing problems associated with an evaluation of computer security. However, a higher level of specificity is being sought in contrast to the first workshop's general coverage. The objective of these efforts is the development and publication of a Federal Information Processing Standard (FIPS) Guideline on the subject.

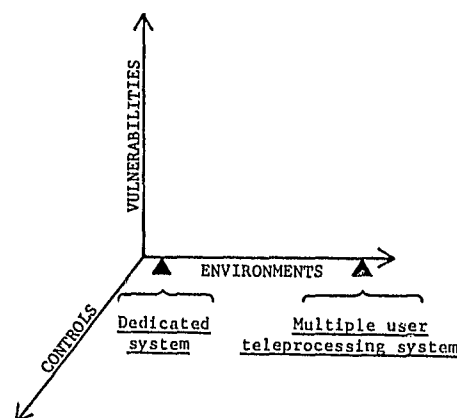
Computer Security Audit Defined

For the purpose of this workshop, a computer security audit is defined as an independent evaluation of the controls employed to ensure (1) the accuracy and reliability of the data maintained on or generated by an automated data processing system, (2) the appropriate protection of the organization's information assets (including hardware, software, and data) from all significant anticipated threats or hazards, and (3) the operational reliability and performance assurance of all components of the automated data processing system.

TASKING

General Information

It is generally agreed that computer security is a function of the environment in which a computer system operates. Using this as a basic premise, computer security can be viewed as a three dimensional problem with the environments, associated vulnerabilities, and controls representing the various problem components.



Security of a dedicated system operating in a benign environment is primarily dependent upon administrative, procedural, and physical controls, whereas a teleprocessing system is subject to additional vulnerabilities thereby requiring incremental controls primarily of a technical nature. In order to simplify the task of the various sessions, a worst-case environment should be considered in addressing their topic area--a multiple-user teleprocessing environment. Other environments may be considered if time permits.

Each session is asked to identify system vulnerabilities from the vantage point of its topic area without regard to the risk of exploitation. The risk of exploitation of any given system vulnerability is dependent upon a number of factors such as the sensitivity of the data which may be targeted, complexity of the exploitation problem, etc. Risk analysis is the subject of other ongoing development and outside the scope of this workshop.

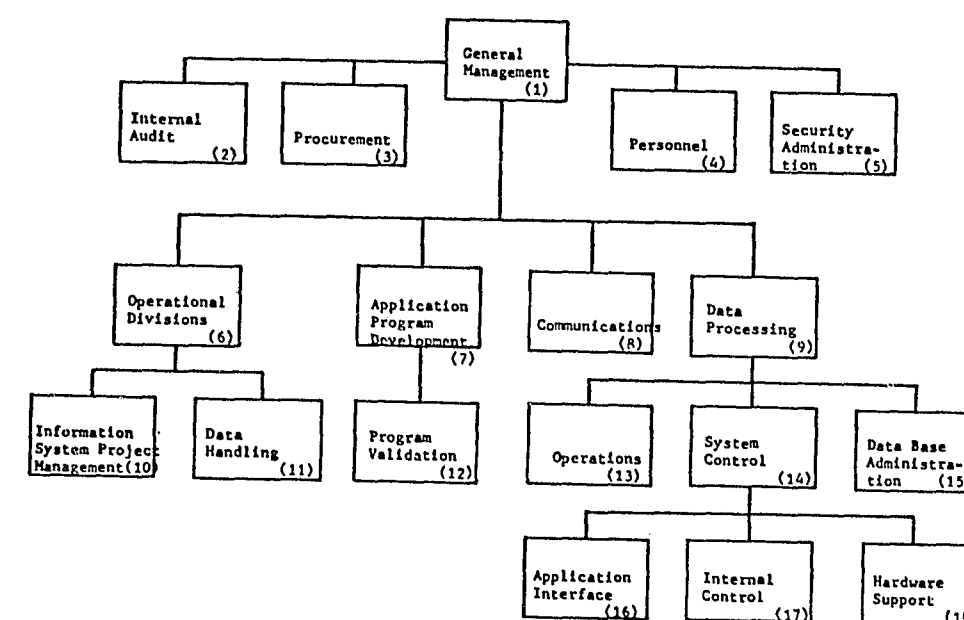
Along with the identification of the various system vulnerabilities, each session is asked to identify those controls which will deter the possible exploitation of each vulnerability and/or detect an actual or attempted exploitation. In this connection, any control which will increase the work factor of an attempt to exploit a vulnerability should be considered. The qualitative effectiveness of any given control should be assessed and results indicated. The cost of implementation, in resources, should also be considered and an order of magnitude assigned.

The end result of the work of each session will be a paper identifying and commenting on the various controls as viewed from their topic's perspective. This effort, together with the results of the first workshop, will be used to develop the FIPS Guidelines on Audit and Evaluation of Computer Security. It is intended that the Guidelines provide the Federal agencies with a general approach to an effective evaluation of computer security together with a matrix for each of several specific system environments which will identify specific system vulnerabilities and related controls. The Guidelines will provide the information necessary for use as a tool in the development of a detailed security audit program tailored to a specific system under evaluation.

Management Sessions

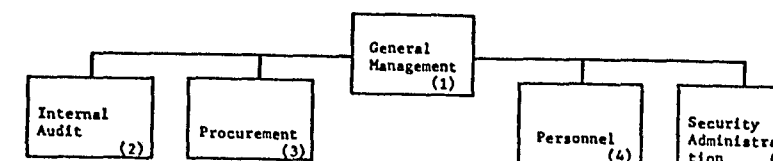
Management at all levels plays an integral part in any effective security program. Therefore, a major segment of the workshop is devoted to this aspect of computer security evaluation. The NBS Special Publication 500-25, "An Analysis of Security Safeguards for Detecting and Preventing Intentional Computer Misuse," was used as a departure point in the development of the following model of an organizational structure to be used in the deliberations on this subject.

A MODEL FOR CATEGORIZING SYSTEM VULNERABILITIES AND MANAGERIAL CONTROLS ACCORDING TO RESPONSIBLE ORGANIZATIONAL UNITS



The model has been divided among three sessions in order to reduce the scope of any one session to a workable segment. Each session is asked to view system vulnerabilities from the managerial level indicated and to identify those controls that can be exercised from that level. The following are the managerial sessions and the segment of the model to be addressed by each.

Session 1. Managerial and Organizational Vulnerabilities and Controls -- Staff Level



Category Definitions.

(1) General Management--This element includes those individuals or functions whose primary responsibility is the management and administration of the agency. This element is responsible for establishing policy and ensuring that adequate resources and line management support is provided to carry out the agency's mission. (Note: It is generally agreed that the absence of top management involvement results in a lack of a) appropriate organizational structure and policies, and b) planning and procedures necessary for the funding, development and implementation of an effective security program.)

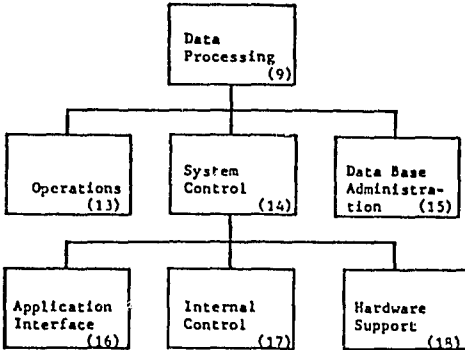
(2) Internal Audit--An independent appraisal activity within an organization for the review of operations as a service to management. It is a managerial control which functions by measuring and evaluating the effectiveness of other controls. The responsibility of this element includes verification and evaluation of controls, standards, and data processing results.

(3) Procurement--This staff element is responsible for ensuring that all contracts, including those involving software and hardware, are properly specified to minimize the potential for loss resulting from automating information systems.

(4) Personnel--This staff element is responsible for maintaining required personal information on employees, as well as providing the official guidelines describing the policy of the agency regarding hiring and firing criteria, background investigations, etc.

(5) Security Administration--This staff element is responsible for developing overall policy and monitoring, on a continuing basis, the overall effectiveness of the agency's security program. A separate Security Administration function may be practical only in large organizations. In smaller organizations, the function may be combined with other functions, but should be independent of data processing operations in any case.

Session 2. Managerial and Organizational Vulnerabilities and Controls --
Line Level-Data Processing



Category Definitions (cont'd)

(9) Data Processing--This element includes the management and operation of all computer equipment, personnel and facilities to meet the agency's data processing requirements.

(13) Operations--This subelement of Data Processing is responsible for the day-to-day operation of all computer equipment. It is also responsible for media control and backup, transport, and storage.

(14) System Control--This subelement of Data Processing is responsible for ensuring the integrity of the operating system and the environment in which applications programs execute. It has three components: Application Interface, Internal Control, and Hardware Support.

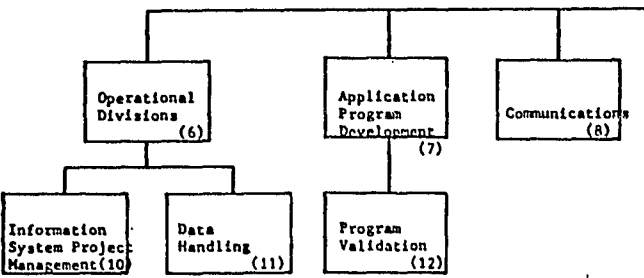
(15) Data Base Administration--This subelement of Data Processing is responsible for the successful management and control of the data bases necessary to support the information processing system. Data base management systems are used to support this subelement.

(16) Application Interface--This component of System Control is responsible for specifying application systems program standards and ensuring that all application programs are adequately tested and documented prior to being placed in the production files.

(17) Internal Control--This component of System Control is responsible for cataloging all internal controls available and ensuring that operational application system controls are in place and working. In addition, this component ensures that the operating system has adequate internal controls and is properly maintained.

(18) Hardware Support--This component of System Control is responsible for ensuring that hardware maintenance is performed in an appropriate and reliable manner. In addition, this component is responsible for the acquisition, planning and maintenance of any hardware required to support security safeguards.

Session 3. Managerial and Organizational Vulnerabilities and Controls --
Line Level General



Category Definitions (cont'd)

(6) Operational Division--Government agencies will have many operational divisions, but conceptually they are all similar from a data processing point of view. Therefore, the model provides for only one operational division. An operational division is an organizational unit responsible for one or more general agency functions. Each division has many branches and sections, each of which, in turn, is responsible for one or more functions relating to the division's mission. Only two of these functions are germane to this model, i.e. Data Handling (11) and Information System Project Management (10).

(7) Application Program Development--For this model, all application program development and support are placed outside of data processing as a separate design activity, even though many agencies place this function within data processing or within their operational divisions. Application program development includes all facets of information system analysis, programming, and testing (in conjunction with the Application Interface (16)) required to develop computer-based systems to support all levels of agency management and operations.

(8) Data Communications--This line element is responsible for the movement of computer-encoded information by means of electrical transmission systems. Data communications is a specialized area of data processing involving such features as terminal devices and special interfacing equipment. Data communications may, in some agencies, be placed as a subelement of a separate communications element or a subelement of the data processing activity.

(10) Information System Project Management--This subelement of the Operational Divisions is responsible and has authority for the successful management of an information system from the users' perspective. This includes ensuring that (a) all user requirements have been identified, appropriately documented, and provided to the system design activity; (b) appropriate user-specified controls are included in the system to assure accurate and timely results; and (c) system performance effectively supports the users' objectives as approved by general management.

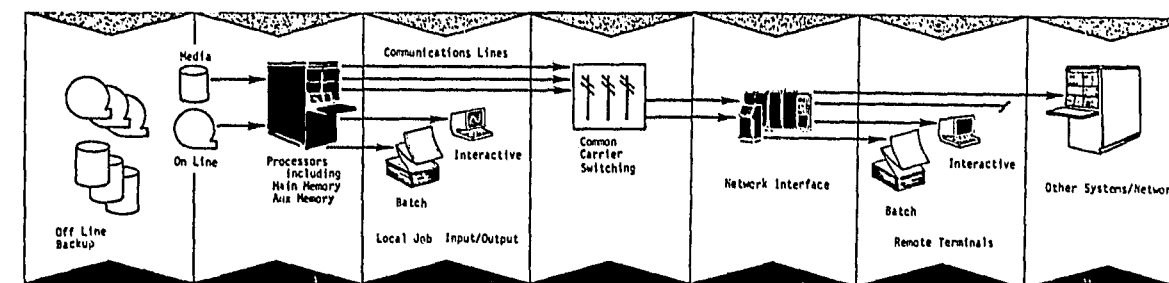
(11) Data Handling--This subelement of the Operational Divisions includes all facets of data preparation, transport to and from input and output devices, and report distribution and storage.

(12) Program Validation--This subelement of Application Program Development is responsible for reviewing, validating, and approving all programs and program changes placed on the system. Where it is impractical to establish a formal and independent test and evaluation group, such as in a small organization or where the programming function is relatively small, mandatory peer review may be employed to provide this program integrity function.

Technical Sessions

The worst-case environment that is being considered at this workshop is that of a multi-user teleprocessing system similar to that illustrated below.

Fig. 1 SYSTEM ELEMENTS



Source: FIPS PUB 41, pp 10 & 11

As with the management sessions, the problem has been divided among several sessions in order to reduce the scope of any one session to a workable segment. Each session is being asked to identify system vulnerabilities related to their topic area and to identify those controls which will deter the possible exploitation of each and/or permit detection of an actual or attempted exploitation.

The following charges the technical sessions with considerations of those segments of a teleprocessing system to be addressed by the respective sessions.

Session 4. Terminals and Remote Peripherals

This session is to consider vulnerabilities inherent in remote processing and the countering controls which may be applied. All types of remote devices should be considered with the exception of those associated with the communications network. Data communications should be viewed as transparent.

Session 5. Communication Components

All modes of data transmission and associated equipment should be considered. Specific vulnerabilities should be identified along with appropriate safeguards, e.g. interception of microwave transmissions, with encryption serving as the countering control.

Session 6. Processors, Operating Systems, and Nearby Peripherals

This session should consider the vulnerabilities associated with the operation and maintenance of the central processor, operating system and hard-wired peripheral devices. Appropriate controls should be considered from two different perspectives: the system design and acquisition phase and the ongoing operational system phase.

Session 7. Applications and Non-Integrated Data Files

This session is to address the vulnerabilities and necessary controls related to applications, application program development and maintenance, and data files where a DBMS is not employed.

Session 8. Data base and Data Base Management Systems

Data base management systems can serve as an important element in the implementation of procedures and safeguards for the protection of information. This session is asked to identify the various vulnerabilities of a data base and inherent in the use of the data base management system. The controls that can be employed to counter the identified vulnerabilities should be addressed.

3. EDITOR'S COMMENTS ON THE SESSIONS AND THE REPORTS

3.1 Definitions of Terms

In the tasking instructions given to the attendees prior to the workshop (see PART I, Sec. 2), the only term defined was "computer security audit." During the course of the workshop and particularly in the period afterwards, during which the Proceedings reports were developed, it became clear to the Co-Chairpersons that a consistent set of definitions for frequently used terms, relevant to computer security audit, would be beneficial. As a consequence, a set of definitions for fifteen frequently used terms was developed by the Co-Chairpersons and circulated for comment, first to interested persons at NBS and GAO, and then to all the attendees of the workshop. Most of the responses were then integrated by the Co-Chairpersons into the initially circulated set of definitions. Some late responses were reviewed and incorporated in the definitions by the Editor alone.

Lack of time prevented an iteration of this comment process. Consequently, the definitions, found in the Glossary in Appendix A, must be viewed as a step in the right direction but not a final consensus view of the attendees. Therefore, though in most instances the definitions will be consistent with the use of these terms in the reports, at times this may not be true. In the Editor's opinion, however the inclusion of this Glossary is still beneficial since most of the definitions were agreed upon to a large degree.

3.2 Materials Distributed at the Workshop

In order to assist the attendees in starting their discussions with a common information base, certain documents were distributed. These included the following NBS publications:

1. "An Analysis of Computer Security Safeguards for Protecting and Preventing Intentional Computer Misuse," Ruder, B., Madden, J. D., Editor-Blanc, R. P., NBS Special Publication 500-25, Jan. 1978.
2. Draft Version of "Additional GAO Audit Standards - Auditing Computer-Based Systems," U. S. General Accounting Office, March 1979.
3. "Automatic Data Processing Risk Assessment," Reed, S. K., NBSIR 77-1228, March 1977. [Now superseded by "Guideline for Automatic Data Processing Risk Analysis," Federal Information Processing Standards Publication (FIPS Pub) 65, Aug. 1, 1979.]
4. NBS Draft Guideline "Security for Computer Applications," June 1978.

Reference 1 had been used by the Co-Chairpersons for developing the model of an organizational structure on which the session topic breakdown was based. Reference 4 was considered particularly useful for Session 7 on "Applications and Non-Integrated Data Files."

A reference list of publications outside of NBS was also distributed and many of these documents brought to the workshop for different sessions to borrow as needed. The reference list is included at the end of this part of the Proceedings.

3.3 Reading the Report

Although the sessions worked independently of one another, except for conversations at refreshment breaks, it is the opinion of the Co-Chairpersons that the workshop is of such a structure that the reader will derive the most benefit from reading the management sections first, due to their general applicability, and the technical sections second, due to their applicability to specific components of the technical environment. Since the management sessions have addressed such organizational units as system control, application interface, data base administration, data handling, application program development, and communications from the management point of view, a second reading of this report would benefit from cross-referencing the related technical sections.

3.4 References

References 1 to 9 below were on the list distributed at the workshop for general background. References 10 to 20 were added in the intervening year as appropriate for this subject. Note that these references are all external to NBS. NBS publishes its own list of computer security publications which is available on request.

REFERENCES ON COMPUTER SECURITY AUDIT--EXTERNAL TO NBS

1. "Systems Auditability and Control Study," Stanford Research Institute Report in Two Volumes, 1977
 - a. Ruder, B., Eason, T. S., See, M. E., Russell, S. H., "Audit Practices"
 - b. Russell, S. H., Eason, T. S., Fitzgerald, J. M., "Control Practices"

2. Mair, W. C., Wood, D. R., Davis, K. W., "Computer Control and Audit," by Touche Ross and Company, published by the Institute of Internal Auditors, 1976
3. "Control Objectives," EDP Auditors Foundation for Education and Research, Hanover Park, Illinois, 1977
4. "Computer Audit Guidelines," by Study Group on Computer Control and Audit Guidelines, Canadian Institute of Chartered Public Accountants, Toronto, Canada, 1975
5. "Computer Control Guidelines," by Study Group on Computer Control and Audit Guidelines, Canadian Institute of Chartered Public Accountants, Toronto, Canada, 1975
6. "Internal Auditing in Federal Agencies," U.S. General Accounting Office, 1974, available through GPO
7. "Standards for Audit of Governmental Organizations, Programs, Activities, and Functions," U.S. General Accounting Office, 1972, GPO SN 2000-00110
8. "Data Security Controls and Procedures--A Philosophy for DP Installations," IBM Publication G 320-5649-0, February 1976
9. FitzGerald, Jerry, "Internal Controls for Computerized Systems," Jerry FitzGerald & Associates, 906 Barkentine Lane, Redwood City, California, 1978.
14. "Auditing Computer-Based Systems," U.S. General Accounting Office, March, 1979, GPO SN-020-000-00174-7
15. "The Auditor's Study and Evaluation of Internal Controls in EDP Systems," by the Computer Services Executive Committee, American Institute of Certified Public Accountants, New York, N.Y., 1977
16. "EDP Controls and Auditing," Porter, W. T., Perry, W. E., Wadsworth Publishing Co., Inc., Belmont, California, 2nd Edition, 1977
17. "A Guide for Studying and Evaluating Internal Accounting Controls," Arthur Anderson & Co., Subject File AA 2880, Item 1, January, 1978
18. "EDP Auditing," Auerbach Publishers Inc., Pennsauken, N.J., 1978
19. "Guide to Accounting Controls," Price Waterhouse & Co., New York, N.Y., 1979, PW 946001-9

PART II: KEYNOTE ADDRESS

DONALD L. SCANTLEBURY
U.S. General Accounting Office



Biographical Sketch

Donald L. Scantlebury is Director of the Financial and General Management Studies Division of the U.S. General Accounting Office. He joined the General Accounting Office staff on October 1, 1956, after several years in public accounting. He served with the Defense Division until October 1964 and with the Field Operations Division as Manager of the Washington Regional Office from October 1964 to June 1971.

Mr. Scantlebury attended Antioch College, Yellow Springs, Ohio, from which he received a bachelor of arts degree in Business Administration, and the Executive Development Program at the University of Michigan. He is a Certified Public Accountant (Iowa and Wisconsin).

He was National President of the Association of Government Accountants for fiscal year 1976-77, and is Chairman of the National Intergovernmental Audit Forum. He is also a member of the American Institute of Certified Public Accountants, the Steering Committee of the Joint Financial Management Improvement Program, and the Executive Committee of the National Council on Governmental Accounting.

He has received achievement awards from the Association of Government Accountants, and the General Accounting Office, including GAO's highest award, the Comptroller General's Award.

He is the author of several articles on accounting and auditing subjects which have appeared in various professional publications.

Keynote Address
Proceedings of the Invitational Workshop on Audit
and Evaluation of Computer II:
System Vulnerabilities and Controls

Donald L. Scantlebury

1. INTRODUCTION

I appreciate this opportunity to address this second National Bureau of Standards Workshop on Audit and Evaluation of Computer Security and I thank Zella Ruthberg and Bob McKenzie for inviting me. I think this is a very important subject and I would like to stress this importance right at the outset.

In the General Accounting Office we have observed an acceleration of concern in the computer security area. We have been heavily involved in promoting this concern--beginning in 1976 with our reports on computer-related crimes, on automated decisionmaking, and on physical security. Since then there has been a lot of talk and even some action:

- (1) The new transmittal memo to Circular A-71 calls for increased protection for Federal computers.
- (2) Senator Ribicoff's investigations and publications will most likely result in the passage of S-1766 or some other computer crime law during the next Congress.
- (3) The recommendations of the Electronic Funds Transfer Commission are being implemented slowly, bit by bit, in amendments to existing legislation. Some of these, designed to give consumers needed protection, imply better security measures by providers and regulators of services.

As one of the Commissioners of the Electronic Funds Transfer Commission (EFTC), I had a chance to see firsthand the present and proposed volumes of money transfers, and also what the vulnerabilities are. We even had some chilling demonstrations of these vulnerabilities by personnel from a government security agency. In my opinion, the Commission did not respond adequately to these vulnerabilities. They felt that because no threats could be documented, no action was needed. We pointed out that crime will inevitably go where the vulnerabilities are--if the potential "take" is big enough.

The famous bank robber, Willie Sutton, was asked why he robbed banks. He is supposed to have replied, "Because that is where the money is." He knew where the money was and he knew enough of the bank's vulnerabilities to be able to get to the money. Today, the same people--the banks--still have the money, but today, they use computers to handle money transactions and that has opened up a whole new group of vulnerabilities. These vulnerabilities are shared by Government and businesses as well as by banks.

As I said, the EFTC did not pay enough attention to the security problem, even though some of you here gave excellent testimony in the hearings. Today we have another chance to attack this issue. Our combined efforts will produce a report and, ultimately, definitive guidelines. I think the report itself will be highly useful in illuminating this subject for action by all concerned.

2. COMPUTERS HAVE BROUGHT NEED
FOR NEW SECURITY CONTROLS

As most of you know, I am an accountant by trade and I tend to view the world with an accountant's eyes. What I have seen during my career is a great change in the way business and Government handle their assets without nearly as much change in the control systems management uses to protect its assets.

2.1 Old Controls are Obsolete

Let me just give you a little of my perspective on this problem. The Venetians are said to have developed double entry bookkeeping in the 15th Century. That was an important step in the development of the systems that management employs to safeguard its valuables. It was not the first step. Even with single entry bookkeeping, there were control mechanisms which managers employed. The controls were developed gradually over hundreds of years until we had a rather effective system that provided reasonable protection of assets in those organizations that followed the system. Let me just cite a few of the tenets of that system:

--All checks should be signed by two people.

--Those who sign checks should examine invoices and supporting documents to see that the goods were received and the prices correct.

--Those who open mail should prepare lists of checks and deposit them immediately. They should have no access to accounting records; particularly records of receivables.

--The bank statement should be reconciled by someone who cannot write checks or receive cash.

Those are a few. There are dozens more.

2.2 New Controls are Needed

Then came the age of the computer. It changed the way transactions and recordkeeping were handled and made many of the controls we accountants cherished obsolete or no longer practical. Let us look at what has happened. What about the good old control of having two people sign checks? Do we still honor that in the Federal Government? Not exactly. If you have a Government check on you, you will note it has only one signature and that is a printed one. Chances are it is signed by Henry Eades, who is the Treasury's disbursing officer. If he tried to sign all the checks that are issued with his signature, he probably could not do it in several lifetimes.

Not only do we not have the checks signed by someone, in the more advanced systems we do not even get the documents together in one place. Transactions in which the order for goods, the invoice from the vendor, and the receiving ticket for the goods are each physically retained in different cities are not uncommon. The match is made by computer.

Assets do not include only cash and inventories either. Many organizations have sensitive information in their computers that they must protect from outsiders. Also, the morality of our times and the computer's ability to assimilate data have made it necessary to protect individuals' privacy. Therefore, the computer has given us a challenge. We, who are concerned with protecting the assets of our organizations--cash, inventories, secret data, private data, and so forth--have to accomplish in a few years what it took hundreds of years to develop before. Moreover, we have to do it in an environment in which we are shooting at a moving target. What I mean by that is that the capabilities of the equipment are changing so rapidly that what works today may not work tomorrow.

As I learned on the Electronic Funds Transfer Commission, we will soon have a paperless system for transfers of funds. The traditional controls will not work for such a system. We have got to have new ones and ones that work. That is the real security challenge before us.

3. PURPOSE OF THIS WORKSHOP

Now I want to discuss this workshop a little. Since there are eight sessions, each of us will be dealing with one of the topics and I thought it might be useful to get a perspective on what all eight will be doing.

To start with, what is different about this workshop compared to the one we had before? First, we are a little older and I hope a little smarter. Second, Workshop One was organized around ten functional topic areas--Internal Audit Standards, Qualifications and Training, etc. Workshop Two is organized around responsible organizational units and technical components. This workshop is expected to give specific information on vulnerabilities and controls from which FIPS guidelines can be framed.

Vulnerabilities can include non-dollar items, such as business-interruption factors, so we need to keep a generalized view of assets to be protected. Vulnerabilities can be overcome by the exercise of controls for prevention and by good tracking and detection procedures. We are here to provide NBS with our best thinking on this subject.

To continue, since each of us has been concerned with only one of the sessions, I thought I would give a brief overview of what each session will cover. To get that data, we contacted each of the session chairmen. My investigation of Session 1 was particularly easy because I am the Chairman of that one.

3.1 Session 1. Managerial and Organizational Vulnerabilities and Controls - Staff Level

In preparing for Session 1, I found that the title "Staff Level" is a misnomer. We are concerned with more than "Staff." The model published in the tasking document shows that this section involved the five top boxes in an organization chart. This includes general management and its supporting staff functions. The challenge of our session is to identify and assess those vulnerabilities which should be addressed directly by top management and how they should address them. Also, we need to set a framework by which top management can determine whether the lower levels of management are doing what they need to do to see that proper controls are established and maintained--in other words--a feedback system.

3.2 Session 2. Managerial and Organizational Vulnerabilities and Controls - Line Level Data Processing

In session 2 you will be considering the management and operation of processing itself. This is an area that has had much attention from both data processors and auditors. Usually each group has considered problems and solutions from their own perspectives. Here, at this workshop, is a real opportunity for both groups to "get it all together" by working jointly to develop meaningful standards.

The central data processing organization is a junction of many diverse management functions: operating system control, data base administration, hardware support, internal control, applications interface, and standards. This is a grand central station of activity with high technical content. The challenge I see here is to keep the workshop directed to the management level rather than to the technical level.

3.3 Session 3. Managerial and Organizational Vulnerabilities and Controls - Line Level

Session 3 concentrates on Line Level General. The organization chart in the model shows three major components in this area: Operational Divisions, Application Program Development, and Communications. I do not know where the organization chart came from, but this area is a major bite out of any organization. Consider that it includes the operational divisions of, say, a major corporation or a large government agency. Here is a challenge of size as well as a technical challenge.

In the many years of looking at controls, many specific procedures have been developed. With the difficulties imposed by new technology, we need to step back, to get away from the specifics, and to look at the broader picture. Can we approach this area in a more general way by first determining the objectives of controls? If so, we might then define areas of vulnerability and be able to apply classes of controls appropriate to them. This would then be followed by the specific techniques and procedures. The challenge is to address the diverse operational situations with objectives, principles, and standards so that all concerned can see what we are driving at when a specific control procedure is called for.

The three sessions I just commented on are supposed to take care of the management portions of the security program. Now we come to the technical sessions.

3.4 Session 4. Terminals and Remote Peripherals

Terminals come with many different attributes. They are dumb or smart, local or remote printer or cathode ray tube. Some are used by the clerical personnel with no computer background; some are used by sophisticated COMPUTERNIKS to alter operating systems that cost millions of dollars to develop. The terminal may be adjacent to the computer or it may be on a space-satellite miles away.

We have a major security challenge here primarily because of the remoteness aspect. What makes this such a difficult security problem? I think it is because we do not have a closed physical entity to control. We have undefined physical space and undefined personnel who might be able to get access to the system. Also, the "remote peripheral" included in this session's title might be a computer system with capabilities beyond those of the one we are trying to protect. So the challenge is: Can we control access by an unknown person, located in an unknown place, and having unknown levels of skill and resources?

3.5 Session 5. Communications Components

This session on communication components includes "all modes of data transmission and associated equipment." Developments in the area are proceeding at a faster pace than even the computer business itself. Major providers have announced new services and protocols that give us a moving target to aim at. If our work here is to have value beyond a few years, we will have to consider the following factors:

- rapid technological change,
- rapid growth in the number of networks,
- rapid growth in the varieties of networks, and
- growth in the volume of traffic per network.

If that is not enough of a challenge, let us not forget what is probably the most important aspect for control--the people involved. People are often considered to be the weakest link in a security system. Here in the network environment we have what appears to be an automatic operation. While communications systems may be automatic, think about all the skilled engineers and technicians required to monitor and maintain them. What added vulnerabilities do they pose?

3.6 Session 6. Processors, Operating Systems, and Nearby Peripherals

Session 6 is supposed to consider the vulnerabilities associated with the operation and maintenance of the central processor, operating systems, and "nearby" peripherals. (I have heard of "plug-compatible" peripherals but "nearby" is a new term. The workshop tasking document also referred to these same peripherals as "hard-wired," which made me wonder if any peripherals are "soft-wired." But these semantic peculiarities are nothing compared to the real problem.)

I cannot think of a more esoteric computer security area than the operating systems and the central processor. This is "where the action is" for the experts. When we hear about people "penetrating" the system or "crashing it," it is usually the operating system that is being talked about. Moreover, this area is getting the heavy attention from those organizations that have funds to spend for security research. The problems are complex and they have not been solved despite the work to date.

There is a reasonably high level of security obtainable in dedicated systems--at a price. But when you talk about multi-user, multi-purpose systems, with multiple security levels, that is a different ball game and there are few professional players. Many of those few are here today.

It has been stated that "there would be no security problem if software people knew how to do their job." That expresses a real challenge to the software community. To the auditing community, I see a different challenge. Let us assume that the software will not get

better soon. Now what can we do about people and other controls that will keep software from being a major factor in the vulnerability equation.

3.7 Session 7. Application and Non-Integrated Data Files

This session is intended to exclude data base management systems. This exclusion leaves everything else in the way of application systems and associated data files. As in our other sessions, there is certainly the challenge of the future here. Also, there is a body of knowledge in the development of secure application systems. But consider that there are thousands and thousands of existing systems in operation today that do not come up to the standards we will be setting. It seems to me that our challenge here is to consider the system now in place as we plan our future efforts. The existing systems should be considered in two ways: first in the present environment, and then in a changing environment.

Anything you do with computer software usually turns out to be an expensive proposition. And if you think that the problem here is limited to the application systems, think a little more. The computer operating systems may be affected by the applications; and, conversely, applications controls are often affected by the operating system. I do not know how deeply this session can address this part of the problem but the large investment in existing systems certainly calls for our earnest attention.

3.8 Session 8. Data Bases and Data Base Management Systems

This, the last session of the workshop, sounds like a very well-defined area with clear limits or boundaries. In a sense it is, but this does not mean it is simple. The state of the art in Data Base Management Systems finds us coming to grips with a variety of structures. There are network, hierarchical, and relational structures with proliferations of access methods, linkages, relationships, and file inversions. There is no universal form for these, and theory and standards are sparse.

The challenge to this session has two major aspects. We must deal with the problem of variety, as in most of the other sessions. The other aspect has to do with auditor independence. When an auditor uses a query language to draw information from a data base, he is dependent on the system structure. Should the auditor have an independent access path to data? I phrased this as a question rather than a position; but let me leave no doubt that auditability is, in itself, a control and we have to provide for auditability in all our workshop sessions.

4. Challenge to the Workshop

The subjects of DBMS and auditor independence provide a convenient place to move our thinking away from the individual sessions and towards the workshop as a whole.

A special characteristic of a data base--practically a definition--is the non-redundant nature of data elements. Ideally in such a system you store information only once and yet you can use the information for multiple purposes. This also means that if you lose any information from the data base, it is really gone. So backup provisions are all important. Thus, vulnerabilities and controls are intertwined with the more general security and audit considerations such as backup and auditor independence. This means that we must address the specifics of this workshop with proper regard for the larger context.

This workshop is a means to an end. The end is safe and sound systems--secure from fraud, and misuse, and accidental error.

Our challenges are extensive, but I believe we can meet them.

- We have a worthy subject.
- We have the right people.
- We have a good location.
- So, now is the time to go to work.

PART III: SESSION 1

MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS -- STAFF LEVEL

Chairperson: Donald L. Scantlebury
U.S. General Accounting Office

Participants:

Robert P. Blanc, Recorder
National Bureau of Standards

Howard R. Davia
General Services Administration

David M. Harris
Lilly and Harris, CPA

Bryan B. Mitchell
Dept. of Health, Education, and Welfare

Frank S. Sato
Department of Defense

Joseph A. Sickon
Department of Commerce



From left to right: Frank S. Sato, Bryan B. Mitchell, Howard R. Davia, Donald L. Scantlebury, Joseph A. Sickon, Robert P. Blanc, David M. Harris.

Note: Titles and addresses of attendees can be found in Appendix B.

EDITOR'S NOTES

DONALD L. SCANTLEBURY

Donald L. Scantlebury is Director of the Financial and General Management Studies Division of the U.S. General Accounting Office. He joined the General Accounting Office staff on October 1, 1956, after several years in public accounting. He served with the Defense Division until October 1964 and with the Field Operations Division as Manager of the Washington Regional Office from October 1964 to June 1971.

Mr. Scantlebury attended Antioch College, Yellow Springs, Ohio, from which he received a bachelor of arts degree in Business Administration, and the Executive Development Program at the University of Michigan. He is a Certified Public Accountant (Iowa and Wisconsin).

He was National President of the Association of Government Accountants for fiscal year 1976-77, and is Chairman of the National Intergovernmental Audit Forum. He is also a member of the American Institute of Certified Public Accountants, the Steering Committee of the Joint Financial Management Improvement Program, and the Executive Committee of the National Council on Governmental Accounting.

He has received achievement awards from the Association of Government Accountants, and the General Accounting Office, including GAO's highest award, the Comptroller General's Award.

He is the author of several articles on accounting and auditing subjects which have appeared in various professional publications.

THE CHARGE TO THE GROUP

This group was originally asked to address the managerial and organizational vulnerabilities and controls at the staff level and to assume there exist staff level units for Internal Audit, Procurement, Personnel, and Security Administration (see the complete charge given to this group in PART I, section 2). The subject was instead broadened by the group to address the responsibilities and duties of Top Management and its relation to its supporting staff and line level functions. The results of these deliberations are thus able to provide a framework within which all of the managerial and technical concerns of the other sessions can be viewed.

The report that follows is the consensus view of this session.

Managerial and Organizational Vulnerabilities And Controls - Staff Level

D. L. Scantlebury, Robert Blanc, Howard Davia,
David Harris, Bryan Mitchell, Frank Sato, Joseph Sickon

1. INTRODUCTION

1.1 General

Top management in any organization is responsible for setting policies, procedures, and standards that will promote efficiency, economy, and effectiveness in all of an organization's activities. Of course, top management will delegate the necessary authority for implementing much of its responsibility to staff and line management. One function that is very important and which must remain as a top concern to the highest level in the organization is data processing. Top management is responsible for establishing physical, administrative, and technical safeguards for its automated data processing systems. Those safeguards must be adequate to protect the assets and the data the systems contain.

1.2 Importance of computer security

In most Federal agencies, the data in the computer falls into one of several categories:

- Financial information.
- General operating information of a management, administrative, and technical nature.
- Information affecting the security of the United States.
- Private information on U.S. citizens.
- Other sensitive information, such as that which could result in competitive harm to Government contractors if made public.

This information must be protected against unauthorized access and use. For example, unauthorized access to accounting data must be protected against because the Government can be defrauded if the data is used to improperly appropriate Government funds or other assets. Because much of the data in computer systems is important to the country and to the operation of the Government, its protection merits top-level consideration.

1.3 Top management responsibilities

The workshop was to decide which responsibilities top management must assume to assure proper security of its data processing systems and which duties it can delegate. The consensus of workshop members was that top management must (1) provide for an organizational structure to assess the vulnerabilities of, and provide effective controls over, its data processing systems, (2) establish policy and control standards which promote secure, well-controlled systems, (3) allocate adequate resources to provide controls and periodically test them, and (4) require periodic reports on security. Top management can delegate authority for establishing security procedures and for performing related duties but may not relinquish ultimate responsibility.

Although top management is responsible for computer security, implementation is substantially at the operating level. Accordingly, top management must make management at the operating level aware of this duty.

2. ORGANIZATIONAL STRUCTURE

2.1 General

Top management should provide for an organizational structure that will provide security controls. This includes assigning designated duties to appropriate operating-level managers and staff officers. The following chart shows top management's responsibilities and the assignment of duties recommended by the workshop members.

2.2 Designate a responsible official

Top management within an agency should designate a principal assistant, an assistant secretary or equivalent, to be responsible for data processing systems. The duties of this person should include all aspects of data processing, including system security.

The assistant secretary should be required to:

- Establish procedures and assign duties to staff and line management for implementing top management's policies and control standards.
- Operate the agency's data processing systems security program.
- See that the sensitivity and vulnerabilities of data processing applications and installations are periodically evaluated.
- Report on security to top management.

In summary, the assistant secretary should be responsible for establishing a control process to assure that appropriate administrative, physical, and technical safeguards are incorporated into all data processing systems.

2.3 Establish an automated information systems committee

Top management should establish an automated information systems committee comprising the assistant secretary, as chairman, and the heads of principal organizational components or their designees. The committee would make policy decisions, provide direction on an agencywide basis, and monitor actions taken by various bureaus or divisions. The committee should be responsible for monitoring and evaluating the adequacy of the organization's security controls and reporting to top management on the need for any changes.

2.4 Assign audit responsibilities

Top management should require its internal auditors to periodically assess the adequacy of controls and security safeguards for its data processing systems. The auditors should evaluate proposed systems at critical stages in development, test prescribed controls of operating systems to see that they are functioning, and assess the physical safeguards of existing facilities. The National Bureau of Standards document "Audit and Evaluation of Computer Security," NBS Special Publication 500-19, October 1977, elaborates on the auditors' duties.

2.5 Assign responsibility for personnel security checks

Top management should require a security check of individuals participating in the design, operation, and maintenance of its data processing systems. (See section 3.5.) It is appropriate to assign responsibility for screening to the agency personnel office.

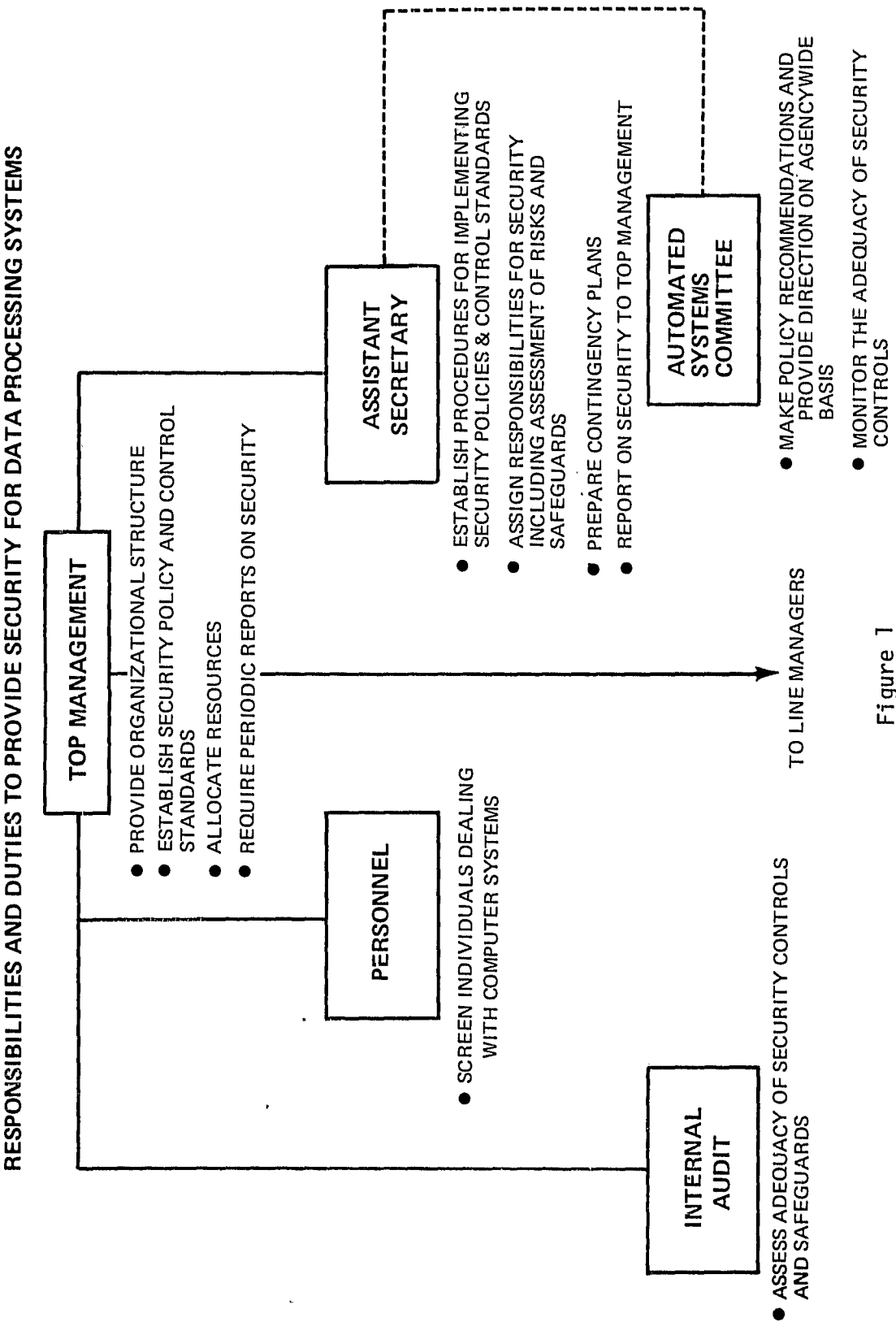


Figure 1

3. POLICY AND CONTROL STANDARDS

3.1 General

Top management should establish the overall policy for protecting the organization's data processing systems and should specify the control standards to be employed. It should require an effective system of controls and secure the funding necessary to establish and maintain such controls.

3.2 Assessing security safeguards

Top management must establish policies to assure periodic assessment of the security of its data processing systems. The policies should provide for assessing the sensitivity of each computer application and the vulnerability of each computer installation and related communication systems.

To effectively distribute security resources according to amount of risk, top management should adopt the risk-management concept for assessing the vulnerability of its data processing systems. That is, the investment of security resources should be based on a formal assessment of the resources to be protected, the controls that are presently in place, and any gaps in security. Perfect security is generally regarded as unattainable. Accordingly, risk analysis, as advocated by the National Bureau of Standards in its publication "Guidelines for Automated Data Processing Physical Security and Risk Management," has considerable merit. Risk management is required by the Office of Management and Budget in Circular A-71.

3.3 Establishing control standards

Suitable guidelines for the protection of the integrity of data in data processing systems are identified in "The Auditor's Study and Evaluation of Internal Control in EDP Systems," published by the AICPA. These standards are endorsed by the workshop. In brief, these 19 standards are as follows.

1. Functions between the EDP departments and users should be segregated.
2. Persons within the EDP department should not be allowed to originate or authorize transactions, have custody over non-EDP assets, and originate master file changes.
3. Functions within the EDP department must be properly segregated.
4. The procedures for systems design, including the acquisition of software packages, should require active participation by representatives of the users and, when appropriate, the accounting department and internal auditors.
5. Each system should have written specifications which are reviewed and approved by an appropriate level of management and applicable user departments.
6. System testing should be a joint effort of users and EDP personnel and should include both the manual and computerized phases of the system.
7. Final approval should be obtained prior to placing a new system into operation.
8. All master file and transaction file conversions should be controlled to prevent unauthorized changes and to provide accurate and complete results.
9. After a new system has been placed in operation, all program changes should be approved before implementation to determine whether they have been authorized, tested, and documented.
10. Management should require various levels of documentation and formal procedures to define the system at appropriate levels of detail.

11. The control features inherent in the computer hardware, operating system, and other supporting software should be utilized to the maximum extent to provide control over operations and to detect and report hardware malfunctions.

12. Systems software should be subjected to the same control procedures as those applied to the installation of and changes to application programs.

13. Access to program documentation should be limited to those persons who require it to perform their duties.

14. Access to data files and programs should be limited to those individuals authorized to process or maintain particular systems.

15. Access to computer hardware should be limited to authorized individuals.

16. A control function should be responsible for receiving all data to be processed, for ensuring that all data is recorded, for following up on errors detected during processing to see that they are corrected and resubmitted by the proper party, and for verifying the proper distribution of output.

17. A written manual of systems and procedures should be prepared by all computer operations and should provide for management's general or specific authorization to process transactions.

18. Internal auditors or some other independent group within an organization should review and evaluate proposed systems at critical stages of development.

19. On a continuing basis, internal auditors or some other independent group within an organization should review and test computer processing activities.

3.4 Require a plan to implement controls

Top management should require the assistant secretary to develop a plan for implementing the controls. This plan should also contain contingency plans to ensure continuity of operations if a loss should occur. The assistant secretary should be responsible for periodically reporting on implementation of the plan.

3.5 Establish personnel security policies

Top management should require the assistant secretary to establish personnel security policies for those employees who deal with its automated information systems. The policy should provide for screening all individuals participating in the design, operation, or maintenance of computer systems. The level of screening required by these policies should vary from minimal checks to full background investigations commensurate with the sensitivity of the data to be handled and the risk and magnitude of loss or harm that could be caused by the individual.

4. ALLOCATE RESOURCES

4.1 General

Top management must secure and allocate the funds and people needed to enable its policy and control standards to be implemented. It must also secure and allocate the resources to enable its prescribed controls to be periodically tested to determine that they are functioning. Top management must also secure and allocate the resources to periodically make a risk analysis of the security of its data processing systems. Fund and staff allocations to staff and line management should be based on the recommendations of the assistant secretary.

5. REPORT ON SECURITY

5.1 General

Top management should require the assistant secretary to periodically report on security. Among other things, the report should state the results of vulnerability assessments and highlight any potential risks which are not provided full protection.

PART IV: SESSION 2

MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS -- LINE LEVEL DATA PROCESSING

Chairperson: Richard D. Webb
Peat Marwick Mitchell & Co.

Participants:

LT COI Robert Campbell, Recorder
HQDA (DAMI-AMP)

Stanley Jarocki
U.S. Dept. of Interior

Keith O. Dorricott
DeLoitte, Haskins & Sells

Harry Robinson
Metropolitan Life Insurance Co.

Lance Hoffman
George Washington University

Carl Williams
American Can



From left to right: Carl Williams, Keith O. Dorricott, Robert Campbell, Richard D. Webb, (Richard Canning, Coordinator of Sessions), Lance Hoffman, Harry Robinson, Stanley Jarocki.

Note: Titles and addresses of attendees can be found in Appendix B.

EDITOR'S NOTES

RICHARD D. WEBB

Mr. Richard D. Webb is a Manager and Senior Computer Audit Specialist in the Executive office of Peat, Marwick, Mitchell & Co.

He has designed and implemented audit software packages and has been a financial and cost accounting systems consultant. Mr. Webb is a Certified Public Accountant (IL and NY) and a member of the American Institute of Certified Public Accountants where he is a member of the Computer Services Executive Committee. He is also chairman of the Task Force that prepared the forthcoming AICPA guideline "Controls over Using and Changing Computer Programs" and is a member of the "Computer Assisted Audit Techniques Audit Guide" project team. Mr. Webb also chaired the Audit Software Specifications Task Force and was a member of the task forces that produced the audit guides: "The Auditor's Study and Evaluation of Internal Controls in EDP Systems," and "Audits of Service Center Produced Records." He is a member and a former Director and Vice President of the New York Chapter of the EDP Auditor's Association and a member of the New York State Society of CPAs. Mr. Webb received his BS in accounting from the University of Minnesota.

THE CHARGE TO THE GROUP

This group addressed the question of managerial and organizational vulnerabilities and countering controls for the line level unit for Data Processing. [See PART I, Section 2 for the complete charge given to this group.] The functional areas of Operations, System Control, and Data Administration were assumed to exist within this unit and the System Control area was discussed under the three functional subunits: Application Interface, Internal Control, and Hardware Support.

The paper that follows is the consensus view of the group.

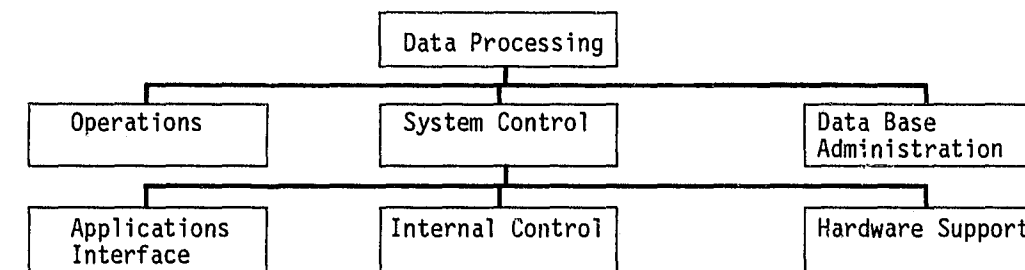
Managerial and Organizational Vulnerabilities and Controls--Line Level--Data Processing

A Consensus Report

Richard D. Webb
Robert P. Campbell
Keith O. Dorricott
Lance Hoffman
Stan Jarocki
Harry Robinson
Carl Williams

I. INTRODUCTION

1.1 MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS--Line Level--Data Processing: The charge given to this session was to produce a structured list of management vulnerabilities, controls, control effectiveness, magnitude of cost, and associated standards that the auditor can use as a basis for preparing a tailored audit program to review security in the organizational areas depicted below:



1.2 Intent of Paper.

The structures of EDP organizations are vastly different across the industry. This provides the auditor with a variety of duties depending upon the organization's structure. In order to provide the auditor with a simplified outlook we have assumed that the above categories represent the working areas of responsibility which can be superimposed on existing organizations. It is not the intent of this group to discuss the interrelationships of an operating environment but rather the elements unique to each working area from a management standpoint.

1.3 Elements for Discussion.

The elements that will be discussed are the control policies (with supporting procedures and techniques for illustration) that management requires in the day-to-day operations of a controlled and reliable data processing environment. The cost and effectiveness of these can only be evaluated in the context of the particular installation being audited and the environment in which it exists.

1.4 Comments.

Efforts of this group were hampered in identification of vulnerabilities and controls by a lack of adequate definition of critical terminology such as threats, vulnerabilities, risk, risk analysis, and risk assessment. Available NBS publications and technical documents (e.g., NBS Special Publication 500-25, . . .) promote confusion by often using these terms interchangeably.

No control policies or procedures were identified for the function "System Control" as these were all dealt with under the three subfunctions.

The function "Data Base Administration" was redefined as "Data Administration" to also apply to file-oriented installations.

2. DATA PROCESSING

2.1 General

There are a number of overall control policies and procedures, which are not dealt with under the subfunctions, that should be in effect in every data processing center in order to have a more secure environment for the data processing operation and at the same time provide the manager with effective means for establishing and maintaining control over the entire operation. Among the more important are the following:

- Organizational Structure and Supervision
- Standards and Procedures
- Personnel Security
- Training
- Hardware and Software Maintenance
- Hardware and Software Quality Assurance
- Equipment Adequacy Monitoring
- Security Management
- Emergency Back-up
- Insurance
- Management Control Reports

2.2 Emergency Back-up and Recovery

The ADP Manager is responsible for establishing procedures that will provide the means for continuing data processing operations at all times despite any emergency. To fulfill this responsibility, procedures for emergency back-up and recovery should be installed, which will include the following:

1. Determination of what data files, documentation, program versions, forms, etc. must be stored in a safe place, off-site, that will provide that means for recovery. For example, this will include special emergency processing procedures.
2. The establishment of procedures for normal replacement of obsolete stored tapes for current ones, as required by the elapse of time, and updating the stored documentation for changes in the trail as such changes are made.
3. Actual live testing to show that a recovery of each transaction trail can be made from the stored data, etc., and that the recovery can be completed within a predetermined time limit. This should be repeated as often as is practical.

4. The establishment of a contingency plan in the event that the computer center is partially or completely destroyed.

2.3 Security Management

Data and other information handled by the data processing equipment and associated telecommunications equipment and systems must be properly safeguarded against unauthorized access, modification, use, and destruction or other denial of use. The safeguarding of such data and information will be accomplished through continuous employment of protective measures designed to ensure the integrity of the data processing capability and of the data bases. These safeguards are based upon a combination of defensive barriers and consist of integrated employment of physical, personnel, communications, emanations, hardware, software, procedural, administrative, and other security techniques.

Among the ADP Manager's security responsibilities, the implementation of the following procedures represent the more important on-going areas:

- A personnel emergency alerting plan and follow-up procedures (such as evacuation, emergency processing teaming, etc.).
- A positive "authorized personnel only" access system for the computer room, library, and other data storage areas.
- A periodic security and safety check utilizing established methodology.
- A procedure for immediate reporting of security violations.
- The location of equipment that receives, processes, and/or supplies output of sensitive data in a secure area suitable to the degree of sensitivity of the data.

2.4 Management and Control Reports

The Manager of ADP should receive periodically a series of reports that provide him information to indicate whether data processing is operating in a normal manner or not. If there are abnormalities in the operations, the reports should pinpoint the area or areas involved and provide the means for initiating remedial action. Procedures for preparing the following types of reports should be in effect:

2.4.1 Production Failure Report. Whenever a production run aborts, regardless of the reason, a Production Failure Report should be prepared by the first person who becomes aware of the abort. This report should initially show all the information available that may have a bearing on the abort. It should also provide information concerning: 1) the investigation undertaken, 2) the identification of the "where" and "how" the error occurred, and 3) recommendations for corrective action. Copies should be disseminated to various areas within data processing to ensure that the cause of the error has been properly identified and that the corrective action recommended will actually correct the problem.

2.4.2 On Time Delivery of Reports. The delivery of routine reports on a scheduled basis is an important aspect of an effective ADP operation. Late reports can result in substantial losses. Users must schedule their workforce on the assumption that routine reports will be delivered on time. Any failure by ADP to meet the schedule can be indicative of problems ranging from minor to very serious. The manager of ADP should routinely receive a "morning report" of those trails where reports were not delivered as per schedule, as well as a weekly or monthly summary of such failures to pinpoint areas where changes must be made.

2.4.3 Performance Monitoring Reports. The usage of all major pieces of equipment should be monitored continuously by the equipment itself where possible, otherwise sampled frequently, e.g., for on-line timesharing and teleprocessing type systems, in order to provide data to indicate whether the usage of the equipment is approaching the maximum potential usage. This may indicate the need for additional equipment if continued growth is anticipated. These monitoring reports provide information regarding machine breakdown and malfunction which is an important determinant of:

- The quality of the maintenance.
- The performance of the equipment itself. (It frequently indicates obsolescence of the equipment and the need for replacement.)
- Higher than normal unit costs to users.
- The quality of the service.

The greater the extent to which a data processing center can avoid maintenance problems occurring at other than scheduled times, the greater will be the security over the entire operation.

2.5 Equipment Acquisition

With the rapidly changing technology and the increasingly greater use of computers and peripheral devices to do more work electronically, it is necessary to have a staff that maintains a current understanding of the state of the art with respect to hardware and associated operating system. Based upon the routine monitoring reports of machine usage and performance, this staff should initiate action to procure additional or replacement equipment as necessary. The specifications should include provisions for the incorporation of security measures to the extent deemed necessary.

2.6 Hardware and Software Assurance

Each time a new piece of equipment and/or operating system software is installed or modified, the equipment or software should be put through a "quality assurance" test. The test should be administered by the hardware support group who have the necessary technical expertise to do this complex work. Generally a quality assurance test requires the use of special input files and test procedures to determine whether the equipment and/or software is performing properly. The special quality assurance test data and special programs are sensitive information that should be given a high degree of protection.

2.7 Training

There should be a training unit and training material to supply the initial training, to be followed later on with on-job training. Besides the training to do a specific job, the training program should emphasize:

- The importance of the data processing operation and the need to follow all security and safety rules and procedures.
- The fact that the data in the files is confidential and should not be disclosed except to authorized personnel.
- The necessity for following exactly all written instructions and procedures. Deviations from established operating procedures should require prior management approval.

The training facility should also include material and facilities for upgrading personnel in the use of new or modified equipment and software, requalifying personnel where necessary, and training in administrative and supervisory methods and techniques.

2.8 Organizational Structure and Supervision

Because of the threat posed by a single person having, or who may later obtain, information on many facets of the data processing operation which may permit him to violate the system, it is necessary to structure the data processing operation so that there is a reasonable separation of duties. Such separation should be designed to prevent one person (and/or one unit) from accessing or working in other parts of the operation which might permit such a violation to occur. There should be procedures and, in some cases, physical barriers to accomplish the following:

1. The library should be a completely protected area with access only to those assigned librarian duties. All data and programs should be stored in the library and only leave the library on valid requisition to run a job. They should be returned to the library immediately upon completion of the job.
2. System software personnel and files should be located in a highly protected area.
3. Data Base Administrator personnel and files should be located in a highly protected area.
4. Printing should be located in an area restricted to authorized personnel only. Sensitive printing should be located in a highly protected area with printer operators specifically trained in maintaining confidentiality of the data.
5. Supervisors, in addition to their normal responsibilities for controlling the work flow through their areas of responsibility, should be trained and kept constantly reminded of the need to be alert to any situation that represents an actual or potential security violation. Any such situation should be immediately reported to management.

2.9 Operating Standards and Procedures

There should be written procedures that control the flow of work. These procedures should include all the operations functions and specify the controls at those points where the work passes from one function to another so that the routine documentation of the flow of the work may be used as the basis for subsequent internal review and audit. There should be detailed instructions for each of the following:

2.9.1 Job Initiation: There should be control documentation for each job that identifies all procedural requirements and is verified by supervisory personnel at all major functional transfer points.

2.9.2 Input/Output. All data entering or leaving the ADP operations area should pass through a control group to verify the accuracy and completeness of the data received or released and conformance with all of the controls established for the input/output operation.

2.9.3 Scheduling and Job Control. There should be procedures governing how to schedule the work, how it is to be processed, and how the job control instructions should be verified.

2.9.4 Logs. System journaling facilities and operating logs record computer activity concerning the runs, errors, restarts, interruptions, and operator interventions. There should be written instructions stating how this data is to be used.

2.9.5 Operating Systems. There should be procedures describing how a computer should be brought up and taken down and how the quality assurance check of the operating system is performed.

2.9.6 Application Programs. There should be procedures governing the testing and installation of applications on the Application Program Master File.

2.9.7 Machine Operations Standards. Machine operations should be measured against pre-determined standards. These standards can be obtained from the vendor, from outside computer operations, or from experience and should include (but are not limited to) the following:

- Machine up-time/down-time
- Maintenance delays
- Number of malfunctions by type
- Terminal response time, etc.
- Reports on time/late

2.10 Personnel.

The ADP manager is responsible for establishing the policies for selecting candidates to fill the various types of work involved in running ADP operations. Candidate selection should take into consideration not only the skills, intelligence, and experience of the individual but also an evaluation of the background of each person to determine whether or not they represent a security risk.

2.10.1 Continuing Checks. There should also be a program designed to check and/or verify the continuing reliability and loyalty of employees after they are hired. In exercising his personnel security responsibilities, the ADP Manager is expected to utilize the services of various intelligence/counterintelligence, criminal investigative, psychiatric, judicial, and personnel agencies available to him.

2.10.2 Minimizing Discontent. There should be policies which take into consideration working conditions, performance evaluation, position requirements, salary administration, etc., to create an environment wherein each employee believes he is being treated fairly and that there is a realistic career advancement path for him so as to minimize discontent and to prevent a lowering of morale.

2.11 Hardware and Software Maintenance

The ADP Manager is responsible for negotiating maintenance contracts for the hardware and systems software that is being used. In doing so, he should take into consideration the reputation of the maintenance contractor, his ability to respond to emergencies both in terms of parts and personnel in a timeframe that is acceptable, and cost. He should also ensure that the scheduling of maintenance will not be disruptive to operations and, at the same time, will not pose a security threat.

2.11.1 Other Important Considerations are:

- Scheduling of preventive maintenance.
- Policies governing maintenance activities and any restrictions on maintenance personnel with respect to type of activity, time, and place.
- Security clearances for maintenance personnel.
- Establishment of quality assurance procedures to check the hardware after maintenance and the operating system software after changes.

2.12 Insurance

If a decision is reached to obtain insurance as a safeguard against destruction of property, the following factors should be considered in determining how much insurance is needed:

-- The replacement cost of hardware, building, and the remainder of the physical environment that might be destroyed.

-- The value of the applications and systems programs. (This value will generally be far greater than the actual development programming and testing costs incurred since the data processing capability depends on the existence and usage of such programs and reverting back to old procedures will either be impossible or extremely costly. It is thus imperative that all necessary steps be taken to protect the program library through back-up, off-site storage, and other methods.)

-- The special costs incurred in operating at another site and/or with other equipment while the destroyed site or equipment is being repaired.

3. OPERATIONS

3.1 General

3.1.1 Functional Responsibilities.

The operations area is responsible for the day-to-day operation of all central site computer and communications equipment and unit record equipment and for the management of related media. This includes the following seven functions:

- Data entry for manually received data
- Machine operation
- Library operation
- Machine utilization
- Output handling
- Environmental control, and
- Access control

3.1.2 Control Requirements. Control in this area requires an orderly and stable operation resulting from effective policies and procedures across all of these functions, including:

- Standard written procedures
- Effective supervision
- Preparation and review of activity logs
- Formal acceptance procedures for new software and hardware and for modification thereto

-- Personnel recruitment, training, job descriptions, security clearance, privileges, and evaluation, and

-- Preparation and review of exception reports of control failures.

3.2 Particular Policies and Procedures in Functional Areas.

In addition to the foregoing common policies and procedures, particular policies and procedures are required within each of the functions as set out below.

3.2.1 Data Entry. For all data received manually, either in source document form for conversion to media or directly in media form, procedures should be developed (in conjunction with submitting departments) to ensure that the data is complete, accurate, authorized, not duplicated, and timely. Where data conversion is required, it should be validated to ensure that the above characteristics are retained.

3.2.2 Machine Operation. To ensure that the correct data and programs are used, that records are not improperly destroyed, and that the installation can survive system failures, appropriate policies with respect to the human operation of computer and communications equipment should be established regarding:

-- The content and use of machine operator instructions (e.g., run books).

-- File label checking

-- The use of restart/recovery capabilities

-- The preparation of error logs

-- The use of protective devices (e.g., tape rings)

-- Permissible operator intervention

-- Backup arrangements (and when they should be employed)

-- Rotating and/or dual operators

-- Validation and authorization of remote terminals accessing the system, and

-- Validation procedures to be included in powering up and enabling the computer and communications equipment.

3.2.3 Library Operation. To prevent the loss, destruction, or disabling of files and/or software, appropriate policies with respect to the maintenance of program, data, and job control libraries, and the use and movement of media should be established. These should include:

-- Identification of media

-- Identification of files and programs (to the version level)

-- Retention back-up and destruction

-- Usage logs

-- Cleaning of media

-- Storage and custody of supplies and media, and

-- Disposal of sensitive media remnants.

3.2.4 Machine Utilization. To detect excessive rerun time, excessive machine down-time, unsatisfactory throughput times and/or excessive user charges, appropriate policies should be established regarding:

-- Regular production of pertinent utilization statistics,

-- Approval of job classifications and priorities, and

-- Machine scheduling

3.2.5 Output Handling. To ensure the detection of processing errors in the output and to prevent output from being distributed inappropriately, appropriate policies should be established regarding:

-- The identification of all printed and other output

-- The quality control procedures over output for detection of errors and/or unacceptable conditions

-- Standard procedures for the physical handling (decollating, binding, etc.) and delivery of output and the disposal of remnants (e.g., extraneous copies, carbon)

-- Standard procedures and assigned accountability for handling of negotiable instruments (e.g., signed checks, purchase orders)

3.2.6 Environment Control. To prevent intentional and/or unintentional damage by human attack or natural phenomena, appropriate policies designed to maintain an appropriate physical environment for the equipment, media, and personnel should be established regarding:

-- The use and testing of protective devices (e.g., smoke alarms) and procedures (e.g., guards), and

-- Installation housekeeping procedures.

3.2.7 Access Control. To prevent unauthorized disclosure, retrieval, or modification of data or programs, appropriate policies should be established regarding:

-- Restriction of physical access to the particular equipment, media, and premises to authorized personnel only.

-- Procedures for appropriate and timely reporting of security violations

-- Location and physical security of premises; and

-- Procedures for actions to be taken with respect to persons charged with security violations.

4. DATA ADMINISTRATION

4.1 General

The data administration area is responsible for the successful management and control of the data files and data bases necessary to support the information processing system. This area is supported by data management systems (which are intended to permit access to and retrieval from existing files, usually in response to single applications, reports generation, or simple information retrieval requirements) and/or data base management systems (which are intended to integrate and manage data in a nonredundant structure for processing by multiple applications).

4.2 Access Policies

The data administrator should set policies to ensure that only properly authorized users have access to the various data and programs in the computer system. The following three definitions are important for the controls considered here:

Identification is the process that enables, generally by the use of unique machine-readable names, recognition of users or resources as identical to those previously described to an ADP system.

Authorization is the granting to a user, a program, or a process the right of access.

Authentication is a measure designed to increase protection against fraudulent transmissions by establishing the validity of a transmission, message, station, or originator.

1. It is important that identification, authorization, and authentication are all carried out (either by the data administration area or by other components of the computer system or both).

2. Since it is possible for authorized or unauthorized users to circumvent data administration controls provided by the data base management system and thereby access data and programs either through the underlying operating system or directly, the data administration area must also be cognizant of operating system access controls and flaws and consider compensatory administrative and procedural controls.

3. Even properly identified users should be granted no access to any data or program unless authorized by a specific office or offices. This type of authorization can be handled in several ways: there can be different categories of users; users can be given various levels of access privileges; there can be different types of access (e.g., read, write, append only); specific programs (procedures) can be invoked before determining whether to grant a particular request. File-level access controls will not always suffice; in the light of Privacy Act requirements, field-level controls will often be necessary.

4. Granting, revocation, or alteration of access privileges should be located in a specific office or offices. Manual and computer procedures should be triggered by granting, revocation, or alteration of access privileges; these procedures should include appropriate logging of these transactions. More detail on this is given in paragraph 5 of this section.

5. Privileges to override normal access control should be granted under special circumstances only. These circumstances should be enumerated in writing. Privileges to override are granted by a specific office or offices (which may be different from the office or offices controlling the granting, revocation, and alteration privileges). Logging of these override events should always be done.

4.3 Unauthorized Statistical Disclosure

The Data Base Management System (DBMS) should attempt to deter unauthorized statistical disclosure output by the system. This is difficult to do automatically. It can be done (to some extent) manually. One can also have a program scan the transaction trail for possible statistical attacks. Further information on this problem and some safeguards for it are given in (2).

4.4 Transaction Trail

The data administration area keeps trails of all transactions for backup, recovery, and audit purposes. When appropriate, the trail should be encrypted. Also when appropriate, the data itself should be encrypted. When encryption is necessary for other than national defense information, use of the NBS Data Encryption Standard (DES) is mandatory for US Government civilian agencies or contractors. Typical information included in the transaction trail is:

- Nature of the event
 - "Logon" and "logoff"
 - Update and inquiry transaction
 - Opening and closing of files
- Identification of all involved elements
 - People
 - Devices
 - Software
 - Data (indexes, directories, files, etc.)
- Information about the event
 - Time and date
 - Success or failure
 - Authorization status of involved elements
 - Transaction numbers
 - System response
 - Addresses of items updated
 - Contents of items/records before and after creation, update, and deletion
 - Programs used
 - Results of compatibility and parameter checks
 - Procedural violations

4.4.1 Automated Journals. For automated journals, due consideration must be given to the problems of archiving extensive data for what may be prolonged periods, even years. The ability to easily off-load voluminous journal data, to condense it as much as possible, and to on-load easily the same data for inspection much later in time, perhaps on a different machine complex, is important. These capabilities may also be required for other reasons--especially for internal and external auditing activities.

4.5 Data Checks and Integrity

The data administration area is the caretaker of the data definitions which are agreed upon by the users of the system. The data administration area is responsible for backup and recovery software for programs and data and for the custodial care of programs and data as supplied to it with no loss or alteration.

5. APPLICATIONS INTERFACE

5.1 General

This component of system control deals with the specification of application systems programming standards and ensuring that all applications programs are adequately tested and documented prior to being placed in the production files. The following statement of controls assumes, for purposes of illustration, that the systems development cycle is composed of the following phases:

- Project definition
- System design
- Detailed design and programming
- System testing
- Conversion

5.2 System Development Controls

Strict control of the system development process is essential to development of secure applications.

1. The more sensitive the application, the more detailed system development controls should be.
2. Project management techniques will ensure systematic (e.g., modular) development of sensitive software, and formal control review/independent acceptance of individual modules prior to system integration.
3. System documentation will be developed concurrent with system development.
4. Security controls will be implemented and enforced commensurate with the sensitivity of the software being developed.
5. Special protection should be extended to all documentation collected and developed which reveals the logic, methodology, procedural aspects, or vulnerabilities of sensitive operations.
6. All vulnerabilities identified during system analysis and design will be brought to the attention of designated management personnel for evaluation of risk, according to formal written procedures.

5.3 Project Definition

User statements of requirement must contain statements of security and control objectives. Each statement of requirement will be reviewed for determination of sensitivity of potential applications, depth of controls required, and general nature of security needs.

1. Prior to undertaking a software development project, both the user and system development organizations will mutually agree upon the project organization, development plan, and level of effort obligations of both parties. Management of the user organization will be advised in detail of the development life cycle, potential resource costs, management decision points, and level of effort required of the user organization.

2. A detailed risk analysis (e.g., identification of threat, vulnerabilities, and relative risks associated with automated operations) will precede the initiation of a software development project.

3. All proposed applications will be reviewed for consistency with applicable laws and regulations (e.g., Privacy Act, Freedom of Information Act).

4. The systems analysis task will document existing security controls and vulnerabilities.

5.4 System Design

Detailed statements of system control and security requirements will be developed and approved by both the user and system development organizations prior to commencing detailed design.

1. User and developer will identify and agree upon input, processing and output controls, audit trail, error control, and file security requirements.
2. During design review and prior to approval/freezing of the design, the user and developer will attest to the adequacy of system security controls.
3. Responsibilities for the initiation, review, and authorization of input transactions will be established.
4. Controls will ensure that all transactions received are entered and processed.

5.5 Detailed Design and Programming

Specific statements of security requirements for data bases will be developed and implemented in the data base design. Highly sensitive data will be identified and special security controls designed to ensure restricted access to that data (e.g., file/record/field lockout, relational prohibitions).

1. Highly sensitive processes and the data upon which they operate will be identified, isolated, and provided special safeguards during detailed design.
2. The adequacy of security controls will be reviewed prior to commencement of formal coding.
3. Coding which operates upon sensitive data or performs sensitive processes will be subjected to special review and verification.
4. Coding intended to provide control and security compliance information will be subjected to special review and attestation as to completeness and accuracy prior to formal testing.
5. Test data will be developed and designed so as to stress both operational and security controls.

5.6 System Testing

All operational control and security features of software will be fully tested and stressed. Tests will identify systems' response to abnormal, unusual, and improbable circumstances.

1. The operation of sensitive processes and integrity of sensitive data will be specially reviewed and verified.
2. Under no circumstances will "live" data be used for system testing.

5.7 Conversion

Conversion of existing files, tables, and data structures will be accomplished under appropriately controlled conditions.

1. All software associated with the performance of conversion operations will undergo the same quality assurance and acceptance as other production software.
2. The integrity of operational data will be verified upon completion of conversion operations.

6. INTERNAL CONTROL

6.1 General

This component of system control is responsible for cataloging all internal controls available and ensuring that operational applications system controls are in place and working. Additionally, this component ensures that the systems software has adequate internal controls and is properly maintained. This includes data communications. This function will be operating the controls established by the application interface function.

6.2 Control Policies

Application systems will contain program to program internal controls. These controls will be verified by the internal control functions.

1. The safeguarding of all documentation supporting applications and systems programs and their interaction is vital to effective security. Protection should be extended to all documentation revealing the logic methodology, or procedural aspects of system operation. This includes but is not limited to:

- Software development documents
- Debug routines and output
- Master control software
- Operating instructions
- Documentation pertaining to software or system errors and flaws

2. The system software, if modified, will be documented as to the intent of the modifications and properly tested prior to installation.

3. All control security features of the operating system software will be cataloged and appropriately applied to ensure control of all application software.

4. No systems software will be modified without approval of the Manager, Data Processing.

5. Operations personnel will be properly trained in the functions of both the application and systems software prior to being accepted as productional.

6. Utilities will be controlled as all other application programs. Each usage will be documented and the input/output placed under the required controls.

7. All production programs will contain version/level controls. Verification of version/level controls will be done by internal control.

8. There will be clearly defined reports which utilize journaling files to detect all unauthorized accesses to the system.

9. A systems acceptance test will be performed by the internal control function prior to the system being placed in production.

6.3 Control Procedures

Program to program controls will consist of record count control and, where appropriate, dollar total controls. Control techniques to be utilized will be documented as per standards and the procedure will be executed as part of the system/program acceptance criteria.

1. Systems documentation and operating documentation will be reviewed and accepted by operations (internal control). Acceptance criteria will be established. Once accepted, all modifications must undergo the same acceptance procedure.

2. Prior to installation, the systems software will be documented as per standards. Modifications will be documented and undergo an acceptance procedure similar to that of applications software.

3. Control/security functions of the systems software will be clearly outlined and installation standards for utilization of same by applications will be produced.

4. Reports will be defined by the internal controls and operations functions.

5. Acceptance procedures will be defined by the internal control and operations functions.

7. HARDWARE SUPPORT

7.1 General

This component of system control is responsible for ensuring that hardware maintenance is performed in an appropriate and reliable manner. In addition, this component is responsible for providing technical expertise for the planning, acquisition, and maintenance of any hardware required to support security safeguards and a reliable operation. It should be noted that hardware support may be Government personnel only, Government and contractor personnel, or contractor personnel only, and that the following discussion covers this broad spectrum.

7.2 Introduction

In this section the auditor will be given the control policies and their associated broad based procedures and techniques necessary for overall hardware support. The control policies for hardware support will cover central site hardware, communications hardware, remote processing hardware, off-line hardware, maintenance personnel, and administrative procedures. Also, the policies will describe where appropriate managerial information is required to ensure that a secure and reliable system is being maintained and that the appropriate safeguards for these actions are being planned for, acquired, and implemented.

7.3 Control Policies

There should be an agreement between vendor and the buyer/user that all system hardware failures that may occur and which may not be detected, e.g., a logic failure that gives the appearance of being operative but when exercised does not stop the intended action (e.g., a defective read/write interlock) or have the appropriate redundancy, will be cataloged and given to the buyer/user.

1. The hardware support personnel will establish and maintain a configuration management and control program. A formal facility profile will be developed and certified by the data processing manager. The purpose of this profile will be to accurately describe or diagram the physical facilities, equipment locations and relationships, and other operating characteristics of the data processing center. This profile will include architectural drawings or diagrams of physical facilities, computer center floor plans, equipment interface diagrams, communications schematics, and wiring diagrams. The facility profile should be maintained and safeguarded by the appropriate official, with no changes allowed to the physical, electronic, or electrical configuration without prior coordination and approval.

2. Determine the criticality of various hardware components according to the needs of the applications that are being run on the system. This should result in an optimal configuration being established to ensure that operations will be able to process data in a timely and efficient manner. Test procedures for these critical components should also be established.

3. Determine the optimum scope and schedule of preventive maintenance; arrange for ongoing supervision to reduce failures to an acceptable level. As a rule, provisions for preventive maintenance will follow a mutually agreed upon schedule between maintenance personnel and the data processing center.

4. Establish and maintain a system incident reporting system. The forms will contain full information including the time of day, system status, tasks and jobs in the system, diagnostic messages, availability of memory dumps, and the like. This information will be properly routed and acted upon. In the case of hardware failure, a record of all repair actions will be maintained and safeguarded.

5. Depending on the size of the installation, one should take into consideration the need for spare parts and full-time maintenance personnel. This will be dependent on the need for immediate repairs as opposed to maintenance which can be acquired by other means in a longer timeframe.

6. An accurate record of all hardware changes should be maintained. Included in this process will be the need to validate these changes as they occur and ensure that they are compatible with current versions of the software and documentation of the system. For example, certain features of the hardware--hardware access checks--may be compatible only with certain releases of the operating system and only when the system is generated properly.

7. The hardware support personnel should be involved in the initial stages of the proposal development to decide on hardware maintenance procedures, checks, and needs. Input at this stage is important to the successful operation of the center during and after installation.

8. Establish a policy for assuring that outside maintenance personnel are properly identified and controlled when in the data processing center. The amount of control will depend on the sensitivity of the operations ongoing in the data processing center and standards of access established for that particular center.

9. Test procedures and their documentation which exercise the security features of the system should be safeguarded. These procedures and their descriptions could disclose sensitive information to persons intent upon penetrating the system.

10. Hardware support should be cognizant of recent advances in the art of system monitoring techniques and equipment that will aid in the detection of system failures, performance evaluation, and analysis of components for proper configuration management. These may take the form of hardware monitors, communications lines and equipment analysis, hardware logic analyzers, etc., which will aid in the continual evaluations of the system and the performance of the equipment. They should provide the technical expertise on these matters to appropriate functional organizations for planning and acquisition. Once these devices are acquired, hardware support should be instrumental in installing them, assuring their quality control, and maintaining them throughout the life of the system.

11. Review the maintenance contract and mean time to fix (MTTF) for all EDP and communications equipment. Determine from where the maintenance is dispatched, and determine if tests can be made from a remote site.

12. Ensure that there are adequate recovery facilities and/or capabilities for loss of key pieces of hardware and loss of various communication circuit/lines.

13. Establish a policy that Federal standards and guidelines are observed in the maintenance of hardware. This is especially true in regard to FPMR 32.7, RP-1, July 1978; and FIPS PUB 31.

8. BIBLIOGRAPHY

8.1 FitzGerald, Jerry, "Internal Controls for Computerized Systems," Jerry FitzGerald & Associates, 506 Barkentine Lane, Redwood City, California, 1978.

8.2 Hoffman, Lance, "Modern Methods for Computer Security and Privacy," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

8.3 Institute of Internal Auditors, "Systems Auditability and Control Study," Stanford Research Institute, 1977.

8.4 US Army Regulation 380-380, "Automated Systems Security," Department of Army, October 1977.

8.5 US National Bureau of Standards, Federal Information Processing Standards Publication 31, "Guidelines for Automatic Data Processing, Physical Security, and Risk Management," June 1974.

8.6 Canadian Institute of Chartered Accountants, "Computer Control and Audit Guidelines," Toronto, Canada, 1975.

PART V: SESSION 3

MANAGERIAL AND ORGANIZATIONAL VULNERABILITIES AND CONTROLS -- LINE LEVEL
GENERAL

Chairperson: Richard J. Gultinan
Arthur Andersen & Co.

Participants:

Keagel Davis
Touche Ross & Co.

Kenneth A. Pollock
U.S. General Accounting Office

Gerald E. Meyers
CNA Insurance

Darryl Poole
American Can

Eric J. Novotny, Recorder
Computer Resource Controls

Steven J. Ross
Irving Trust Company



From left to right: Gerald E. Meyers, Steve Ross, Keagel Davis, Kenneth A. Pollock, Richard J. Gultinan, Darryl Poole, Eric J. Novotny.

Note: Titles and addresses of attendees can be found in Appendix B.

EDITOR'S NOTES

RICHARD J. GULTINAN

Mr. Richard J. Gultinan is a partner of Arthur Andersen & Co. He is Director of Computer Auditing for the New York Metropolitan Area of the Firm. He joined Arthur Andersen & Co. in 1951, spent several years in auditing and then transferred to the Administrative Services Division where he had extensive experience in the design and implementation of data processing systems, as well as in providing support to the Audit Department in connection with auditing such systems. Mr. Gultinan rejoined the Audit Department in 1976 in connection with his assignment of responsibilities in Computer Auditing.

Mr. Gultinan graduated from the University of Notre Dame. He is a C.P.A. and a member of the American Institute of C.P.A.'s, The New York State Society of C.P.A.'s, and the National Association of Accountants.

Mr. Gultinan recently concluded a term of three years as Chairperson of the AICPA Computer Services Executive Committee. Prior to assuming the senior AICPA committee role, he served for four years as Chairperson of its Computer Audit Subcommittee. He is also active with the New York State Society of C.P.A.'s where he is a member of the Professional Conduct Committee, and past Chairperson of the Management Services Committee and its Data Processing Subcommittee. He was a member of the Advisory Committee to the Institute of Internal Auditors project on Systems Auditability and Control and is presently a member of the Business Panel on White Collar Crime of the Chamber of Commerce of the U.S.

THE CHARGE GIVEN TO THE GROUP

This group was asked to address all line level organizational units other than Data Processing. [See PART I, Section 2 for the complete charge given to this group.] That included consideration of (1) Operational Divisions (with Information System Project Management and Data Handling as its subunits of interest here); (2) Application Program Development; and (3) Data Communications. It should be noted that the panel came up with additional subunits of its own choosing.

The report that follows is the consensus view of this session.

Managerial and Organizational Vulnerabilities and Controls Line Level - General

Richard J. Gultinan, Keagle W. Davis, Gerald E. Meyers, Eric J. Novotny,
Kenneth A. Pollock, Darryl V. Poole, Steven J. Ross

1. INTRODUCTION

This panel addressed itself to the identification of appropriate managerial control objectives and potential system vulnerabilities and exposures (risks) relating to computer system activities. We considered system vulnerabilities and potential control and security techniques from a management perspective for each of several organizational elements. We considered this appropriate because management must be considered to be accountable for system actions and misactions. Further, management must take an overall responsibility to assure the existence of reasonable standards for system control and security, and undertake responsibility to assure that all system user personnel understand their responsibilities and duties in complying with such standards.

2. APPROACH

In undertaking our task, we assumed a "worst case" environmental situation employing multiple use teleprocessing systems. Further, we restricted our considerations to those organizational elements of the hypothetical governmental agency structure designed by National Bureau of Standards which were assigned to us, and which included the following:

1. Operational divisions
2. Information systems project management
3. Data handling
4. Application program development
5. Data communications
6. Program validation

2.1 Operational Divisions - (Section 4)

An Operational Division is considered to be an agency organizational unit responsible for one or more general agency functions. We assumed a management unit relating to the Information System Project Management and Data Handling Activities.

2.2 Information System Project Management - (Section 5)

This sub-element of an Operational Division was assumed to be responsible and have authority for the successful management of the hypothetical information system from the user's perspective. Its activities were considered to include ensuring that (a) all user requirements have been identified, (b) appropriate user specified controls are included in the system to assure accurate and timely results, and (c) system performance effectively supports the user's objectives as approved by general management.

2.3 Data Handling - (Section 6)

This sub-element was considered to be responsible for all facets of data preparation, transport to and from input and output devices, and report distribution and storage.

2.4 Application Program Development - (Section 7)

Under the assumed hypothetical agency organizational structure, all application program development and support are situated outside of data processing as a separate design activity even though many agencies place this function within data processing or within their organizational units. This element was, therefore, considered to include all facets of information systems analysis, programming and testing required to develop computer-based systems to support all levels of agency management operations. (As indicated by the hypothetical agency structure, we assumed that establishing standards for programming, testing and documentation of applications was not the responsibility of this group, but that we were required to assure that such standards were complied with.)

2.5 Data Communications - (Section 8)

This element was considered as responsible for the movement of computer-encoded information by means of data transmission systems. We concentrated on questions of control and security of system transactions. We assumed that considerations relating to specific communication system hardware and transmission path components would be covered by the technical panel groups established for that purpose.

2.6 Program Validation

This sub-element was identified as responsible for reviewing, validating, and approving all programs and program changes placed on the system. From a management perspective, we consider its activities as an inherent part of both the Information Systems Project Management (Section 5) and Application Program Development (Section 7) elements and did not retain it as a discrete and separate organizational group for purposes of this report.

3. CONCLUSIONS

The panel believes that the objectives of a system of controls and the risks associated with a failure to achieve them can and should be identified. These are discussed in Sections 4 through 8 which follow. While these objectives and risks are not necessarily new, the panel believes they need to be reemphasized in all entities. In addition, the panel believes the following overriding control objectives must receive a higher level of attention than has been the case in the past.

1. Management has the ultimate responsibility for system controls. Therefore, they must have an appropriate comprehension and understanding of controls and where they can break down.

2. Users have a non-negotiable responsibility for the controls in their systems. Clear definition of user controls and user understanding of those controls are essential to overall system controlability.
3. Short and long-term planning and budgeting within a properly designed organization structure is a key internal control. Without them, the entity will not be able to recognize, accept or manage the changes that take place within the entity.
4. An appropriate systems development methodology which requires active management and user participation and which results in a documented structure of systems controls is essential to managing and maintaining the structure of control and to auditability.

The panel was also in agreement that the definition, in this report, of standards of control practice at the procedure or technique level was impractical, as the procedures used to effect control in a given environment will be selected based on a variety of internal and external factors. We, therefore, have taken the approach of identifying, in Sections 4 through 8, objectives and risks for each assigned organizational element and, in addition, have suggested some illustrative control procedures which might be used by management to achieve its objectives.

The panel emphasizes that there exist a great number of procedures which might be used in any given situation to satisfy management's control objectives and to prevent, detect and correct errors. We have, therefore, made no attempt to identify all such procedures or to evaluate them in either a cost benefit or effectiveness sense. Instead, the reader is referred to the specific control techniques which are already covered, in sufficient detail, in presently available professional literature, such as:

Data Processing Control Practices Report - Systems Auditability
and Control Study
Institute of Internal Auditors,
Altamonte Springs, Florida, 1977

Computer Control Guidelines,
Canadian Institute of Chartered Accountants,
Toronto, Ontario, Canada, 1970

The Auditor's Study and Evaluation of Internal Control in EDP Systems,
American Institute of Certified Public Accountants,
New York, N. Y., 1977

Computer Control and Audit
Mair, Wood and Davis, Institute of Internal Auditors,
Altamonte Springs, Florida, 1976

EDP Controls and Auditing,
Porter and Perry, Wadsworth Publishing Company, Inc.
Belmont, California, 1977

Control Objectives
EDP Auditors Foundation
Hanover Park, Illinois, 1977

Certain of the areas addressed by this panel, particularly with respect to the Operational Divisions element, such as planning, budgeting, etc. are very general in nature. However, we submit that these activities are basic and fundamental to the existence of strong management control over the line activities covered, and consequently to control and security of the computer related activities addressed by this report.

4. OPERATIONAL DIVISIONS

For purposes of this report, Operational Divisions are considered responsible for agency management activities relating to the Information Systems Project Management and Data Handling functions. They are responsible to top agency management for assuring that procedures have been designed and are utilized to provide reasonable assurance as to the control and security of data and data processing systems activities within those organizational units.

4.1 Objective - Long-Range Planning

4.1.1 Nature of the Objective: Procedures should exist for the preparation and periodic updating of a structured long-range plan, in order to assure, to the extent practicable, that current priorities as to organizational and systems activities are established in the light of projected agency directions. In preparing the long-range plan, its developers should give consideration to such internal and external factors as:

1. Agency and divisional goals and objectives.
2. Anticipated impact of technological developments.
3. Changing regulatory requirements.
4. Requirements for compliance with anticipated legislative actions relating to areas such as privacy, computer systems security, etc.
5. Management information requirements.

The long-range planning procedure should provide for periodic reviews of progress as to both the plan itself and the attainment of the established goals. The starting point for the plan should be clearly identified in order to provide a basis for periodic progress reviews.

4.1.2 Risks: In a general sense, the risk associated with an absence of long-range planning is that current activities, staffing, priorities, hardware and software acquisitions, etc., may be disorganized and misdirected. This could have severely negative effects on divisional performance. Examples include:

1. The division may fail to achieve its mission satisfactorily.
2. Agency and divisional resources may be wasted or misused.
3. The division may not be able to comply with regulatory or legislative requirements on a timely basis.
4. Installed systems may become technologically obsolete.
5. Agency and divisional management may not have sufficient information for measuring progress and performance.
6. The division may not be able to manage adequately its activities relating to organizational and systems changes.

4.1.3 Illustrative Control Procedures: A standardized procedure for structured long-range planning should be prepared and documented. It should be communicated to the organization and implemented. Planning responsibilities of all organizational components should be clearly identified at all levels, both as to preparation and periodic updating.

The plan should cover all activities, current and projected, of the organization. It should be consolidated for each organizational unit, for a division and for the agency as a whole. The completed plan should include a clear statement of the anticipated impact of projected changes on all organizational elements, and should be approved by top divisional and agency management.

4.2 Objective 2 - Short-Range Planning (Budgets)

4.2.1 Nature of the Objective: A short-range plan or budget should be developed to set forth the goals and activities of the operational division and each of the sub-elements during the current fiscal period. It should be prepared giving consideration to present status, long-range priorities and directions as established by the long-range plan, and practical and affordable short term progress. It should clearly provide:

1. Appropriate allocation of available financial, personnel and other resources.
2. Recognition of the need for systems and operational modifications and projects to achieve them.
3. Identification of staffing and costs for continuity of current operations and systems.

4.2.2 Risks: The absence of a short-range planning and budgetary process could result in disorganized and non-productive activities in the current fiscal period. Specific negative results could include, among others:

1. The division may experience cost overruns and may exceed its funded expenditure levels.
2. Personnel resources may be inadequate.
3. Personnel and other resources may be non-productive.
4. Systems development activities may be wasteful or duplicative.
5. Progress toward achieving long-range objectives may be unsatisfactory or altogether lacking.
6. Current systems may become obsolete.
7. Current systems may fail due to an inability to satisfy operational requirements.

4.2.3 Illustrative Control Procedures: Control procedures for long-range planning also apply to short-range planning. The short-range plan should provide for a practicable plan, recognizing current constraints as to personnel and funding, and providing, as much as possible, some measure of progress toward achieving long-range goals. Each organizational unit's plan should be reviewed and approved by the next higher level management, as well as by appropriate functional management.

The plan should also provide for measuring current performance, on a period by period basis against the short-range plan. Adequate information should be provided for effective management analysis and control.

4.3 Objective 3 - System Contingency Planning

4.3.1 Nature of the Objective: A formal and structured plan should be established to provide for operational continuity in the event of a major or extended failure of a system or system component. It should recognize the potential for system degradation at various levels and the potential impact upon operations at each such level, and clearly set forth policies and procedures to be followed to minimize such impact and provide for timely system recovery.

4.3.2 Risks: The risk associated with inadequate planning for system failures embodies the inability to discharge operational responsibilities satisfactorily during the period of outage. Certain operations are more critical than others and a failure to maintain highly critical operations could have severe effects upon agency and divisional financial and operational results. Examples include:

1. A severe and extensive system outage could result in a loss of effective management control of its operations, and in a worst case situation, failure of an agency mission.
2. Progress toward achieving short-range goals could be severely impeded.
3. Excessive personnel, outside contractor and other costs may be experienced.
4. Information and other assets may be destroyed or lost.
5. There may be increased exposure to fraud.
6. Personnel may be idle and, consequently, non-productive.

4.3.3 Illustrative Control Procedures: A disaster or contingency plan should be established to recognize the potential for system interruptions and to provide formal and structured instructions and facilities for maintaining critical operations and providing for timely recovery. It should be specific, well documented, and should be explained to and understood by all appropriate personnel. Further, it should be approved by management. Documentation of the plan should be retained in secure, but readily available, on-site and off-site locations.

The level of detail in which the plan is prepared will vary, to a degree, depending upon the nature of, and criticality of agency operations. It should include, for example, provision for:

1. Levels of criticality of applications and activities.
2. Establishment of priorities.
3. Pre-disaster reduction of vulnerabilities.
4. Strategies for recovery.
5. Identification of people to be notified.
6. Identification of available hardware and software backup, including application programs and operating systems.
7. Procedures for recovery or replacement of data files.
8. Identification of need for and sources of forms.
9. Documentation of insurance coverage.

Depending upon the criticality of operations, it is often advisable to test the plan on a periodic basis to assure that it is practicable and understood by all appropriate personnel.

4.4 Objective 4 - Organizational Communications

4.4.1 Nature of the Objective: This objective recognizes that the best intentions of management can fail to be achieved due to a lack of organization, dissemination and understanding of agency policies, procedures and responsibilities relating to security and control. Conversely, management at all levels is hindered in the effective discharge of its responsibilities if not kept fully informed on the results of day to day activities through appropriate and timely reporting techniques.

Management's policies relating to transaction and other authorizations, its approved procedures for transaction handling and processing, and its requirements for control and security should be formally and completely documented. Specific responsibilities of personnel at every organizational level should be clearly explained and a structured reporting system should be defined, implemented and maintained.

Disseminated procedures should include identification of the organizational structure, and of upward, downward and lateral unit interrelationships.

4.4.2 Risks: In general, the risk associated with inadequate organizational communication is a misunderstanding of objectives, policies, responsibilities and procedures, as well as an inability to manage day to day activities effectively. Specific results might, for example, include:

1. A failure to achieve management's financial or other objectives due to a lack of understanding of the objectives.
2. Employee frustration caused by a lack of understanding of management's policies and plans.
3. The existence of incompatible or conflicting objectives at various organizational units.
4. A failure to comply with legal or regulatory requirements.
5. A failure to observe established management policies.
6. Weaknesses or gaps in the system of control and security.
7. An inability to monitor current developments.
8. Erroneous data and reports.
9. A misinformed management, and consequently, faulty decisions.

4.4.3 Illustrative Control Procedures: Management's policies and procedures should be documented in detail in official procedures memoranda or a procedures manual. With respect to the Operational Divisions considered by this panel, the policies should include, for example:

1. The organization and effective use of a management steering committee to establish system priorities, consider problems, review progress against planned objectives, consider short term needs, etc.
2. The requirement for user involvement in system design and user approval of all system modifications.
3. Clear identification of authorization requirements and specific exception procedures.
4. Definition of the responsibilities of the various organizational levels.
5. Reporting requirements and schedules.
6. A statement of management priorities.

4.5 Objective 5 - Personnel Administration

4.5.1 Nature of the Objective: Personnel administration procedures should assure that personnel at all organizational levels understand their duties and responsibilities, are adequately trained in their duties, are effectively monitored, and are objectively evaluated as to performance.

4.5.2 Risks: The risks of inadequate personnel administration are many and varied. With respect to those areas considered by this panel, they could include, for example:

1. Low productivity due to inadequately trained or misinformed employees.
2. Employee frustration caused by a lack of understanding of responsibilities.
3. Employee dissatisfaction due to inadequate performance evaluation.
4. Abuse or misuse of agency resources.
5. Violation of data integrity as a result of misunderstanding control and security procedures.
6. Loss of data or system control due to poorly trained employees.
7. Organizational incompetence.
8. Organizational mismanagement.
9. Exposure to fraud.

4.5.3 Illustrative Control Procedures: Management should establish specific policies and practices to be followed in personnel administration. These policies and practices should include, for example:

1. An agency code of conduct relating to all personnel which covers conflict of interest situations, gifts, expense accounts, relationships within and outside the organization, etc.
2. Employee interviewing, screening, hiring and termination practices.
3. Provision for adequate employee training.
4. Provision for rotation of duties, and job enrichment, as appropriate in the circumstances.
5. Adequate separation of employee duties in sensitive data handling areas.
6. An effective performance monitoring system.
7. A fair and objective performance evaluation system.

5. INFORMATION SYSTEM PROJECT MANAGEMENT

This organizational unit is responsible to the Operational Division to assure that all systems are successfully designed, implemented and controlled from a user management perspective. Risks associated with a failure to achieve system control objectives discussed below are considered generally applicable to all such objectives. Consequently, risks and illustrative control procedures will be discussed on an overall basis, rather than identified with a specific objective.

5.1 Objective 1 - User Involvement in System Design Activity

Procedures should exist to assure that users of a planned system are involved, in depth, in all phases of system design and development activities. Their involvement should encompass, for example, such responsibilities as:

1. Definition and identification of input data edits, file, field and report requirements, etc.
2. Proper documentation of such requirements.
3. Effective communication of requirements, in detail, to the system design activity.

5.2 Objective 2 - User Specification of Controls

The system design methodology should assure that control requirements are specified in detail by system users and are included in system design. User responsibilities in this respect include the following, among others:

1. Definition of specific control requirements.
2. Documentation of all such requirements.
3. Effective communication of such requirements to the system design activity.
4. Continuing involvement with system design activity to assure that required controls are implemented properly in the system.

5.3 Objective 3 - Continuing User Satisfaction

Procedures should exist for frequent user monitoring of operational systems performance and control to assure that the system continues to meet its objectives in terms of control and security.

5.4 Objective 4 - User Compliance With External Requirements

In connection with all system related activities, users should undertake to assure compliance with all external system requirements, including:

1. Legal.
2. Regulatory.
3. Interagency and intraagency.
4. Internal and external auditor needs.

5.5 Risks

The overall risk resulting from inadequate user involvement in system design activities is obviously the failure of implemented systems to satisfy agency needs and objectives in a controlled, cost effective and productive manner. Specific risks are many and varied. They include, for example:

1. A failure to satisfy one or more specific user or external requirements.
2. Inadequately controlled systems.
3. Exposure to fraud.
4. Faulty system security.
5. Loss of accountability and the ability to reconcile data.
6. Unnecessary or excessive costs resulting from overdesign of a system or system reports.
7. Loss or misuse of resources.
8. Delayed implementation schedules.

5.6 Illustrative Control Procedures

Among many potential control procedures applicable to Information Systems Project Management, the panel believes the most important include the following:

1. An adequate systems development life cycle or other systems development methodology.
2. Adequate user responsibilities in systems development.
3. Adequate user responsibilities in systems changes and maintenance.
4. Adequately designed and implemented user controls.

5.6.1 System Development Life Cycle or Other Systems Development Methodology: The system development process should be organized into specific phases, such as:

1. Project definition and survey.
2. Preliminary system design.
3. System design.
4. Application software development and system testing.
5. Implementation.
6. Post installation review.

This concept of phased system development is covered, in depth, in available literature.

The requirements of each phase should be clearly spelled out and understood by all involved in the development process. Specific management checkpoints should be established during, and at the end of each phase, to assure that project goals and objectives are being realized and that costs to date, and projected for the future, are within established parameters.

5.6.2 User Involvement in System Development: -- The organization of each system project development team should provide for specific user involvement and participation throughout the development process. User responsibilities should include, for example:

1. Identification and documentation in detail of all user oriented system requirements.
2. Determination of the economic and operational feasibility of the project.
3. Establishment and documentation of project scope and objectives.
4. Project review and sign-off at specific checkpoints.
5. Involvement in system testing, user training and conversion activities.

5.6.3 User Involvement in System Changes and Maintenance: All changes to implemented systems should be approved by user management. User personnel should also participate in the testing process and authorize the implementation of the change.

5.6.4 Design and Implementation of User Controls: Specific control procedures for user personnel should be established, documented, communicated and understood by all involved personnel. These should cover, for example, such areas as:

1. Separation of responsibilities.
2. Authorization levels.
3. Responsibilities for master file changes.
4. Security of data and files.
5. Documentation and auditability requirements.
6. User responsibilities to specify application input, processing and output controls for data processing activity and to assure that implemented systems achieve the required level of control and security.

6. DATA HANDLING

The area of data handling constitutes those organizational entities directly responsible for transformation of external information into machine-usable data,

and vice versa. These duties include data transcription, dissemination, storage and retrieval. The controls in data handling are direct, aimed at the physical integrity of the data and the organizational integrity of the information those data represent.

6.1 Objective 1 - Maintenance of integrity of input data

6.1.1 Nature of the Objective: The integrity of input data should be maintained at all times. The accuracy, completeness and timeliness of the data being processed by the computer determines their usefulness to the organization. In computing, data is a raw material; any loss of data integrity will result in a flawed finished product. Procedures should exist to assure that:

1. All transactions are authorized by the appropriate person or persons.
2. Specific job functions include the ability to authorize certain input data within prescribed limits (dollar amount, geographic area, etc.).
3. All valid transactions are authorized by the system, and conversely, all invalid transactions are rejected so that all input batches are complete and error free.
4. Data are entered correctly and on a timely basis.
5. Edits assure that all data used are correct.
6. In on-line systems, each transaction entered is positively acknowledged to the enterer and is logged by the system to assure the data enterer that the system has accepted each item.
7. At the end of processing, all items are balanced against the day's total master file.
8. There is a transaction trail for all data entered to allow management to be able to recreate the path, both forward and backward, for all items in the system.

6.2 Objective 2 - Correct and timely reporting of exceptions

6.2.1 Nature of the Objective: All exceptional conditions should be reported in a thorough and timely fashion. Since modern data processing systems are constructed on the basis of "management by exception," management can only exercise its function with regard to computerized operations if it is presented with meaningful information, where and when it is most needed. Procedures over exceptions should include the following:

1. Clearly established responsibility for responding to errors which the system identifies and positive actions to correct the exceptional condition.
2. Requirements for suspending transactions in cases in which errors cannot be immediately corrected, and methods for updating the suspense files in order to maintain financial control over errors.
3. Requirements for reconciling input to output once corrections have been made, to assure completeness and maintenance of financial and data controls.
4. Maintenance of accurate records of the number, type, distribution, and concentration of data handling errors to be used to develop statistics useful in identifying systematic and managerial weaknesses.
5. Resolution of exceptional conditions in an expeditious manner in order to avoid delays in error correction which could compound existing problems.

6.3 Objective 3 - Secure information storage, retrieval and use

6.3.1 Nature of the Objective: Information should be stored, retrieved, and used in a secure manner. Data should be protected against malicious and inadvertent destruction, modification and disclosure.

Data, and by extension, information, are organizational assets and as such must be secured against loss. The peculiarities of a data asset necessitate all the awareness of security required for tangible resources, plus a number that are specific to data.

Procedures related to data security should include the following:

1. Identification and categorization of data elements by differing levels of sensitivity.
2. Delineation of individuals authorized to handle data, set forth in the same manner as security over the data.
3. Unique identification of users by the computer in order that their authority to access sensitive data can be verified prior to release of the data.
4. Storage in such a fashion, that data are protected against physical destruction, and are recoverable (or at least recreatable) if destroyed.
5. Design of systems to satisfy all statutory and regulatory requirements, e.g. privacy, nondisclosure, conflict of interest, etc.
6. Design and operation of systems in accordance with accepted organizational policies and practices, to reduce the overall exposure to litigation and statutory sanctions.
7. Monitoring systems regularly and continuously for breaches of security with all such events being responded to immediately.

6.4 Objective 4 - Controlled dissemination and storage of information

6.4.1 Nature of the Objective: The information produced by the system should be disseminated and stored under suitable controls.

Output data may be either the end product of the system, or may be a report of its internal operations. In either case, these data are (or should be) produced for use, and thus should be used for their intended purposes, and only those purposes. Procedures for output data should include the following:

1. Review of all reports by supervisory personnel for reasonableness, accuracy, and exceptional conditions.
2. Designated supervisory responsibility to take appropriate action based on the content of those reports.
3. Distributing all output in a timely fashion.
4. Reconciliation of all output data to the input data entered originally, with checks on any transactions generated internally.
5. Distribution of output data only to those who have a demonstrated need for them, with such need being periodically reviewed and evaluated.
6. Secure storage of information while needed, with the information being properly destroyed when no longer required.

6.5 Risks: The risks associated with failure to manage and control the data handling area properly can expose the entity to many potential problems including,

inaccurate or incorrect input data, improper output information, statutory sanctions, litigation, etc. Examples of risk include the following:

1. Admission of errors into the system in the form of erroneous or duplicated work, or outdated records.
2. Elimination of vital records from the files.
3. Introduction of unauthorized or possibly fraudulent transactions into the system.
4. Alteration, destruction, or disclosure of data in an unauthorized manner.
5. Susceptibility to fraud, statutory or regulatory sanctions, criminal and civil penalties, etc.
6. Production by the system of erroneous or out-of-date reports and other forms of output upon which management may improperly rely.
7. Failure of the organization to accomplish its stated objectives or its mission.
8. Exposure to direct financial loss.
9. Indirect financial loss arising from difficulties in reconstructing financial or other information assets.

6.6 Illustrative Control Procedures: Because the controls over data handling are inherently the controls over data processing in general, they have received considerable scrutiny by writers and researchers in EDP auditing and controls.

These include such as:

1. Verification of input data.
2. Input batching and editing
3. Run-to-run balancing.
4. Reconciliation of output.
5. Secure storage of input and output data.

It is strongly recommended that readers refer to the bibliography in Section 3, preceding, for identification of illustrative control procedures. Rather than illustrate many specific controls, the panel noted that it is critically important for the reader to realize that a suitable control structure must be developed and implemented as part of the systems development cycle.

7. APPLICATION PROGRAM DEVELOPMENT

This organizational unit is responsible to general agency management for the management and performance of all systems design, programming and testing activities required to support agency data processing requirements. Specific standards and policies relating to the techniques and approaches to be followed in discharging these responsibilities are expected to be established by the application interface unit of data processing. This unit is responsible for assuring compliance with such standards and policies.

7.1 Objective 1 - Program Development Standards

Program development should adhere to established standards for coding and testing methodology, internal controls, documentation, and security.

7.2 Objective 2 - System Development Life Cycle Check-Points

Systems should be reviewed at prescribed check-points in keeping with the system development life cycle (see Section 5.6.1). At these points, both user and auditor approval should be obtained.

7.3 Objective 3 - Coordination with Organizational Plans

The development of an application system should be linked to overall organizational plans.

7.4 Objective 4 - Project Management System Control

All elements of performance in analysis, programming, and testing should be controlled and monitored by a project management system.

7.5 Objective 5 - Testing and Review

All programs should be subjected to testing and review by developers, users, internal auditors and systems validation (quality assurance) prior to implementation.

7.6 Risks

In an overall sense, the risks associated with inadequate control of application program design, programming and testing encompass a failure to satisfy management's information objectives and requirements effectively and productively. Specific risks include:

1. Organizational objectives with regard to resource allocation (general mission, time, money) may not be met.
2. Systems may be uncontrolled and unauditable, and as a result, may introduce errors into organizational records.
3. Systems may not satisfy internal and external requirements.
4. Systems will be difficult to maintain.
5. The organization may be susceptible to business risks such as interruption of operations, competitive disadvantage, and statutory sanctions.
6. Outputs generated by the system may be in error and may result in falacious internal and external reports.

7.7 Illustrative Control Procedures

Again, as in the preceding sections of this report, there are a variety of procedures available for use to satisfy the above objectives, all of which are covered in great detail in available literature. All four procedures outlined in Section 5.6 of this paper, Information System Project Management - Illustrative Control Procedures, also apply in this area. Other specific control procedures might include:

1. The existence of a formal methodology and procedure relating to program testing, including provision for test data preparation and retention.
2. Periodic reporting of progress against plan to senior management.
3. Adequate interface with the quality assurance function to assure compliance with requirements.
4. Adequate interface with internal and external auditors to assure satisfaction with auditability requirements.

8. DATA COMMUNICATIONS

The "worst case" environmental situation we considered implies the use of some one or combination of remote terminals or processing devices transmitting data over communication paths to another processing location. The control objectives and related risks enumerated in Sections 4 through 7 of this paper apply to the communication components of the system as well as any other components, and will not be repeated here. The existence of remote operations and transmission facilities, however, adds new dimensions to risks related to the integrity of data transmitted, and system security and reliability. Of these, certain result from the characteristics of the devices and of the communication paths utilized, and we have assumed that the technical panel groups, established for the purpose, would give adequate attention to these matters. Consequently, we have limited our comments in this section to the three objectives of integrity, security and reliability, extended risks in the teleprocessing environment, and illustrative control techniques.

8.1 Objective 1 - Integrity of Data Transmitted

8.1.1 Nature of the Objective: The use of remote data terminals and communication facilities expands the potential for the introduction of incomplete or erroneous data into a system. All of the control objectives discussed in Sections 6.1 and 6.2 of this paper, and related risks discussed in Section 6.5, apply as well to the remote and communication aspects of the system. In addition, extended procedures should exist to assure that data are not lost or unintentionally altered due to the remoteness of access devices or the physical characteristics of the transmission paths.

8.1.2 Risks: Additional risks resulting from the on-line nature of the system relate to the entry or receipt of erroneous or incomplete data to or from a central system. These additional risks result from several factors:

1. Employees using access devices may not be adequately trained.
2. Input formats may be overly complex.
3. Terminals transmitted to may be out of operation.
4. Communication paths may be interfered with by natural disturbances, such as electrical storms, or by physical problems in some component.
5. Terminal transmissions may interfere with one another.

8.1.3 Illustrative Control Procedures: Management should establish specific policies and procedures to assure data integrity, including, for example, such as:

1. Adequate terminal users manuals.
2. Fixed terminal input formats.
3. Adequate data balancing controls.
4. Message numbering and logging.
5. Centralized control of communication networks, utilizing polling and specific device identification.

8.2 Objective 2 - System Security

8.2.1 Nature of the Objective: Systems utilizing remote access devices and communications facilities can provide heightened opportunities for deliberate misuse of system files and data by both employees and outsiders. The control

objectives discussed in Section 6.3 and 6.4 of this paper, and related risks discussed in Section 6.5, apply as well to the communication aspects of the system. In addition, extended procedures should exist to help assure that the system and system data cannot be deliberately compromised or destroyed by employees or outsiders.

8.2.2 Risks: In an overall sense, the danger of an inadequately secured system is that unauthorized employees or others may gain access to the system. Specific risks include:

1. The system may be exposed to fraud.
2. Sensitive data, master files or programs could be examined or stolen by employees or outsiders.
3. The system or system data could be damaged or destroyed by disgruntled employees or outsiders.
4. Data confidentiality or privacy could be compromised.

8.2.3 Illustrative Control Procedures: Procedures designed to help prevent security breaches could include, for example:

1. Specific identification of users and terminals, under central system control.
2. Location of terminals in a secure physical environment.
3. The use of passwords to authorize system access.
4. System monitoring and logging of access attempts and transmissions.
5. System notification to security personnel of suspicious or unusual network activity.
6. Use of multilevel data file, data and transaction access controls.
7. Use of data encryption methods for highly sensitive transactions.

8.3 Objective 3 - System Reliability

8.3.1 Nature of the Objective: In a multiple use teleprocessing system environment, the potential for system outage or inadvertent destruction of programs or data is heightened due to the geographical dispersion of system components and the effect of natural disturbances on communication facilities. The objective, risks and illustrative control procedures related to system contingency planning, discussed in Section 4.3 of this paper, should be considered carefully in establishing a plan for system backup and recovery. In addition, system response times should be carefully monitored to avoid deterioration.

8.3.2 Risks: Additional risks related to system reliability in the teleprocessing environment include:

1. Terminal maintenance and repair may be disruptive and time consuming.
2. Failure of all or a part of communication facilities may render a system temporarily unusable.
3. System response times may deteriorate due to increased volumes, inadequate human factors at terminal sites, or equipment malfunctions.

8.3.3 Illustrative Control Procedures: Procedures to minimize the extended risks in this area might include:

1. A scheduled preventative maintenance program for remote devices.
2. Development of a plan for temporary voice telephone or other transmission of data during an outage period.

3. Provision of back-up equipment and/or communication facilities in time-critical situations.
4. Network and response time monitoring.

9. SUMMARY - THE CASCADING EFFECT OF MANAGEMENT RISK

Following the panel's identification of the four major areas for systems management concern, we decided to examine the risks we have identified and associated with control objectives. By subjecting our sub-element vulnerabilities and exposures ("risks") to a frequency distribution, we have identified a critical, cascading or "Tier" effect of management exposure.

Four principle risk levels are identified in cascading levels of importance.

9.1 Organization Mission Impacts

These are elements that will directly and negatively affect the unit's performance. 41 incidences of risk are identified in this Tier.

9.2 Information Reliance Impacts

These are elements of information dependency that will render executive decisions null over time. 19 incidences of risk are identified in this Tier.

9.3 Control Disciplines

These are elements of basic systems control subject to compromise, distortion and mismanagement. 8 incidences of risk are identified in this Tier.

9.4 Organization Disciplines

These are elements of basic managerial skill and organization analysis, especially those where a lack of sensitivity and comprehension will negatively impact the unit's mission delivery capacity. 4 incidences of risk are identified in this Tier.






















9.5 Conclusions

We find that if breaches or failures occur in any component of a higher tier, then several effects will follow in subsequent tiers. The importance of this chain effect should not be overlooked. Risks and increases in exposures run both upward and downward in cause and effect.

On our diagram which follows we have identified (across the top) major Control Objectives and (on the vertical side) Risk Tiers. This diagram illustrates several joint observations.

1. Data Handling, at the level of successful management of Tier 1, Organizational & Mission Impacts, is the most critical cluster. 17 incidences of risk are identified in Tier 1 alone for Data Handling.
2. The opportunities to take preventative actions are more available to planning activities within Operational Divisions (19 incidences of risk) and to Information Systems Project Management (9 incidences of risk) than other areas.

FIGURE 1 MANAGERIAL & ORGANIZATIONAL VULNERABILITIES AND CONTROLS-LINE LEVEL GENERAL
(CASCADING EFFECT OF MANAGEMENT RISK: IF-THEN)

- RISKS-  = WILL CAUSE RISK NUMBER.
  = CAUSED BY RISK NUMBER
- ☐ TOTAL FOR SECTION/AREA
- TIER 1: ORGANIZATIONAL & MISSION IMPACTS**
1. Failure or unsatisfactory performance of mission and/or goals; financial loss or business risk; loss of relevant position.
•  2, 3, 4, 5, 6
 2. Loss, alteration or destruction of data/assets/information; defalcation exposure; increased fraud exposure; unauthorized disclosure; erosion of security; ungraceful degradation of controls; violation of data integrity.
•  3, 11  1, 3, 4
 3. Misinformation and erosion of managerial decisions; data integrity and accountability loss of confidence as effects both internal and external reporting requirements; unrecognized flaws and/or systems network violations.
•  2, 6, 10, 11  1, 2, 4
 4. Inability to comply with legal, regulatory, legislative requirements; incursion of statutory and/or judicial penalties, litigation.
•  2, 3, 6  1, 5
 5. Waste, abuse or misuse of personnel, financial, time and/or organizational resources; decreased productivity effectiveness and potential; misallocation.
•  4, 6-10, 11-14, 15-18  1
- VERTICAL FREQUENCY TIER 1*
- TIER 2: INFORMATION RELIANCE IMPACTS**
6. Failure of current systems to satisfy operational requirements & short range goals/objectives and/or other informational needs; declining performance.
•  7-10, 11  1, 3-5, 7, 8
 7. Unsatisfactory reliability or inability to utilize information mechanisms in building toward longer term objectives.
•  6, 8, 9  6, 11
 8. Erosion of organizational confidence, internal as well as external, due to managerial and employee misunderstanding, miscommunication, frustration and dissatisfaction; inability to sustain internal organization communications.
•  6, 9, 10, 11, 13, 14, 16  5, 6, 7, 9, 13, 18
 9. Insufficient basis for either monitoring or measuring performance and/or progress; uncontrolled systems environments; non-auditability of systems environments.
•  8, 14  8, 10, 14
 10. Premature failure or obsolescence of current systems; non-maintainable; unsatisfactory ability to reconstruct in timely and/or cost effective manner; unacceptable abridgement of planning.
•  9, 11  3, 6, 8
- VERTICAL FREQUENCY TIER 2

AREAS OF RESPONSIBILITY												
4. OPERATIONAL DIVISIONS					5. INFOR- MATION SYSTEMS PROJECT MANAGE- MENT	6. DATA HANDLING				7. APPLICA- TION SYSTEMS DEVELOP- MENT	8. DATA COMMUNIC- ATIONS	HORIZON- TAL FREQUENCY
LONG RANGE PLAN- NING	SHORT RANGE BUDGET PLAN- NING	SYS- TEMS CON- TIN- ING	ORGANI- ZATIONAL COMMUNI- CATIONS	PER- SONNEL ADMINIS- TRATION		DATA INPUT INTEG- RITY	EXCEP- TIONAL (RE- PORT- ING) CONDI- TIONS	INFOR- MATION SECUR- ITY MAN- AGE- MENT	INFOR- MATION PROD- UCT AND OUTPUT			
X		X	X		X	X	X	X	X	X		9
		X	X	X	X	X	X	X	X		X	9
			X		X	X	X	X	X	X	X	8
X			X		X		X	X	X	X	X	8
X	X	X			X			X	X			7
3	1	3	4	2	13	5	3	4	5	17	3	41
	X	X			X	X					X	5
	X					X	X			X		4
			X	X	X						X	4
X			X							X		3
	X		X							X		
1	3	1	3	1	9	2	2	1		3	3	19

CONTINUED

1 OF 3

TIER 3: CONTROL DISCIPLINES

11. Impact of systems outage/interruption on operations, mission; or defalcation, degradation and recovery; on effectiveness of managerial control.
 - ▼ 2, 3, 6, 12, 13 ▲ 7, 9, 11
12. Cost overruns (uncontrolled) and exceeding of approved expenditure levels; negative performance.
 - ▼ 1, 5, 13 ▲ 11, 13
13. Unplanned duplication of systems development efforts; fragmentation of authority over standards and controls; costly over design; damaging under design.
 - ▼ 8, 11, 12, 14 ▲ 11, 12, 14
14. Incompatible and/or directly conflicting objectives across varying organizational units; significant and uncontrollable policy deviation.
 - ▼ 5, 8, 9, 13 ▲ 9, 13

VERTICAL FREQUENCY TIER 3

TIER 4: ORGANIZATIONAL DISCIPLINES

15. Premature technical obsolescence of newly installed systems.
 - ▼ 5, 13, ▲ 11, 18
16. Inadequate selection, identification, and assignment of personnel resources.
 - ▼ 17, 18, ▲ 5, 8
17. Inability to correctly identify and adequately manage further organizational, functional environmental, and/or systems changes.
 - ▼ 16, 18, ▲ 16, 18
18. Organizational mismanagement; managerial and staff incompetence; professional obsolescence.
 - ▼ 16, 17, ▲ 15, 16, 17

VERTICAL FREQUENCY TIER 4

VERTICAL FREQUENCY - TOTAL

	X						2
	X	X					2
	X		X				2
		X					2
2	2	1	5	2		1	8

X						1
X						1
X						1
X						1
2	1	1	4			4
6	7	6	8	4	9	5 5 5 5 20 6 6 72

3. Organizational Communications (mission, purpose, intents, facts, policy, events, directives, consistency, etc.) within Operational Divisions is the second most vulnerable managerial responsibility (8 incidences of risk).
4. Impacts affecting the integrity and reliability of a systems environment are influenced by the shorter range or tactical plans, leading us to conclude that long range business and systems plans are necessary to successfully support the shorter range budget process.
5. Although Tier 4, Organizational Disciplines, appears final on the chart and low in numeric value, we observed that if these elements fail in any permutation, the combined effect undermines the application of control disciplines.

Thus, this panel has concluded that it is important to reemphasize the critical, cascading effect of:

1. Management's overall responsibility for controls.
2. User's non-negotiable responsibility for controls in their systems.
3. Short and long term planning and budgeting.
4. An appropriate systems development methodology.

PART VI: SESSION 4

TERMINALS AND REMOTE PERIPHERALS

Chairperson: William H. Murray
IBM Corporation

Participants:

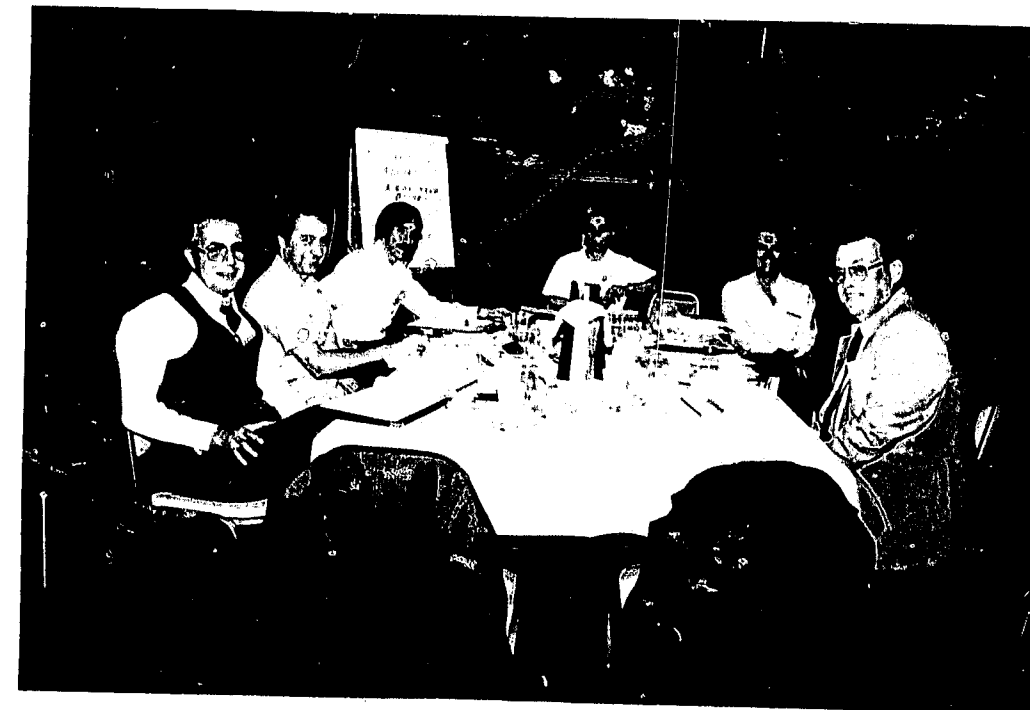
Robert V. Jacobson
International Security Tech. Inc.

George Steffen
Bank Administration Institute

W. Gregory McCormack II
Western Southern Life

Lt. Col. Malcolm L. Worrell
Air Force Audit Agency/ SW

D. V. Stavola
IBM Corporation



From left to right: D. V. Stavola, W. Gregory McCormack II, George Steffen, William H. Murray, Robert V. Jacobson, Malcolm L. Worrell.

Note: Titles and addresses of attendees can be found in Appendix B.

EDITOR'S NOTES

W. H. MURRAY

William Hugh Murray is Senior Marketing Support Administrator in the Data Security Support Programs Department of IBM's Data Processing Division. He is the author of the IBM publication "Data Security Controls and Procedures" and of five IBM training videotapes on data security. He is a contributor to several other IBM publications on data security including "Considerations of Physical Security in a Computer Environment."

He is a frequent speaker on data security topics. National programs on which he has appeared include the AICPA, IIA and the EDP Auditors Association, INFO 76 and Data Comm 77. He has appeared before SHARE and GUIDE in the U.S., SEAS and the Diebold Research Program in Europe.

In 1974, he chaired the Audit Working Group of the "Workshop on Controlled Accessibility in Shared Resource Computer Systems," sponsored jointly by the National Bureau of Standards (NBS) and the Association for Computing Machinery (ACM). In 1977, he chaired the Administrative and Physical Controls session of NBS Invitational Workshop on Audit and Evaluation of Computer Security.

In a previous IBM assignment, Mr. Murray managed the development of the security sub-system for IBM's Advanced Administrative System. This security system permits managers in 400 locations around the world to control the access of 16,000 users to the 900 transactions in 16 sensitive business applications. After ten years of operation this is still considered to be a "state-of-the-art" example of a secure system.

Mr. Murray joined IBM in 1956 as a programmer in the Boardman Road Research Laboratory in Poughkeepsie, New York. He received his Bachelor of Science degree in Business Administration in 1962 from Louisiana State University.

THE CHARGE TO THE GROUP

This session was to consider vulnerabilities inherent in remote processing and the countering controls which may be applied. All types of remote devices were to be considered with the exception of those associated with the communications network. Data communications were to be viewed as transparent. [See PART I, Section 2 for the complete charge given to this group.]

The report that follows is the consensus view of this session.

Terminals and Remote Peripherals

William Hugh Murray

1. TASK AND ASSUMPTIONS

1.1 Task

This session was asked to address the vulnerabilities inherent in remote processing and to recommend the appropriate countering controls. We were asked to consider all types of devices, but to view data communications as transparent. We were asked specifically not to consider the probability of exploitation of a vulnerability. Neither did we consider consequences. For example, in considering the vulnerability of modification of data, we did not consider whether or not anyone was motivated to do it nor what the result might be.

1.2 Audience

We have attempted to present our work in a manner that is useful to an auditor. More specifically we have attempted to present the material so that it will be useful to an auditor at the remote site.

1.3 Useful Life

In an attempt to give this work the longest possible useful life, we have tried to be as independent of any given technology or implementation as possible. We have attempted to view both vulnerabilities and controls in the most general terms. Therefore, specific devices, media, vulnerabilities or controls are considered only as examples or illustrations.

1.4 Limitations

1.4.1 Remote Only: In addition to treating communications as transparent, we elected to consider only those things that are under the direct control of the local (remote) management. Thus, we did not consider application or host system controls.

The auditor is cautioned that a site may be both local and remote for purposes of his audit. To the extent that a site has local applications he will also wish to review its controls as described in the report of the working group on Applications and Non-integrated Data Files.

1.4.2 Terminal Selection: We did not consider the appropriateness of the terminal for the security of the application. We assumed that security was a selection criteria for the terminal. However, the auditor is cautioned that a new application may be added to a preexisting terminal. The availability of the terminal may be the only selection criterion employed.

2. CHARACTERISTICS OF THE REMOTE TERMINAL ENVIRONMENT

2.1 General

In viewing security in the remote terminal environment, the auditor must consider those characteristics of the environment that will influence the selection of appropriate measures. The group identified three such factors. They are application, number of terminals and terminal characteristics.

2.2 Application

The biggest single characteristic influencing the risk of a system is the application or applications. A system that is used exclusively for personal computing will have different control requirements than one being used for business transactions. A system being used for application development may require still different controls. The system that is being used for all three of these may require the most stringent controls of all. The more flexibility or choice that is presented to the end user, the more rigorous must be the controls.

2.3 Quantity of Terminals

The number of terminals in the location will also influence the sensitivity and choice of controls. In general, sensitivity will increase with the number of terminals. Therefore, the auditor should expect to find a more rigorous application of controls in a multiterminal site.

2.4 Terminal Characteristics

A number of characteristics of the device itself were identified which affect sensitivity and the choice of controls.

2.4.1 Portability: Sensitivity was found to increase with the portability of the terminal. Portable terminals are more susceptible to theft. In addition they may be removed to an unsupervised site so as to avoid supervisory control.

2.4.2 Bandwidth: In general, sensitivity can be expected to increase with the bandwidth or character rate of the terminal. For example, it would be easier to mount an exhaustive attack (see paragraph 3.2.2.2, sec.3) upon the host using a paper tape driven terminal than a keyboard driven one.

2.4.3 Storage: Sensitivity will increase with the amount of local storage in the device. (Also see discussion of media, paragraph 3.3.4).

2.4.4 Value: Other things being equal, the vulnerability of a remote site will increase with the value or marketability of the terminal used.

2.4.5 Construction, Modularity and Assembly: The susceptibility of the device to theft or conversion may vary with the way it is built. For example, a nonportable device may be composed of portable modules. The value of a device may be primarily associated with one or two removable cards of chips.

2.4.6 Intelligence: The vulnerability of the host system to an exhaustive attack may be influenced by the intelligence in the remote device. For example, a local processor might be programmed to mount an exhaustive attack (see paragraph 3.2.2.2, sec.3)

2.4.7 Emanations: The susceptibility of the system to the disclosure of sensitive data to eavesdropping varies with the amplitude of signal bearing emanations as a function of the total emanations of the same type.

2.4.8 Media: Vulnerability will vary with the number and types of media supported by the device. In general, sensitivity will increase with the number of types and sensitivity of the media types employed. (See media, paragraph 3.1.3).

3. VULNERABILITIES

3.1 Targets

The group elected to view vulnerabilities of the system primarily in terms of the targets within the system, i.e., in terms of the things which might be vulnerable. Four such targets were identified: 1) data and programs, 2) media (as distinct from the data recorded on it), 3) the terminal or device, and 4) the service or capacity of the system. We believe this list of targets to be complete.

3.1.1 Vulnerabilities of Data and Programs: The group agreed that data was vulnerable to accidental or intentional, but unauthorized modification, destruction or disclosure. We believe this list of vulnerabilities to be complete.

Four characteristics of data were discussed as to their effect on vulnerability. We reached a consensus on three and were unable to agree on the fourth.

3.1.1.1 Location: It appeared to the group that the vulnerability of the data will be influenced by whether it is stored in the host, in the local device or upon external media (because of the assumptions (see paragraph 1.4), only external media at the local (remote) site were treated.). The feeling of the group was that data on external media were most vulnerable to disclosure, but that data stored in the device or host were more vulnerable to modification or destruction. Vulnerability of data on external media will vary with the media (see paragraph 3.1.3). Data in the host were considered to be safer (from vulnerabilities influenced by the remote site) than data in the local device.

3.1.1.2 Form: Natural language data were judged to be more sensitive to disclosure than coded data, while coded data appeared to be more vulnerable to (undetected) modification.

Image data appear to be marginally more vulnerable to disclosure than coded (including text) and less vulnerable to modification.

We also considered the distinction between analog and digital data. However, this distinction appears to exist primarily in communication links and is not relevant to "remote sites".

3.1.1.3 Sensitivity: The vulnerability of the data varies with its inherent sensitivity. This inherent sensitivity is a function of quantity, context, age and degree of analysis.

- o Quantity: The sensitivity of data increases with quantity along an "S" shaped curve, i.e., the sensitivity increases more slowly with quantity for large quantities. This is because large quantities of data start to look like noise (forest and trees effect).

- o Context: The sensitivity of data varies with its context. It tends to increase along an exponential step function with the number of distinct associations such that "employee number" and "salary" taken together are significantly more sensitive than either alone and adding "name" increases the sensitivity by an order of magnitude.

- o Interpretation: The sensitivity of data increases with the degree of analysis or interpretation such that raw data is less sensitive than organized data which is less sensitive than the conclusions which may be drawn which are less sensitive than the plans of action.

- o Age: In general, the sensitivity of data decreases with age. However, there are exceptions.

3.1.1.4 Kind or Use: Three kinds of data were identified. They are user or application data, system or control data, such as security tables, and programs or procedures.

There was considerable discussion in the group as to whether or not to treat data and programs separately or together. The majority felt that programs are substantively different from other data and that it would be misleading to treat them together. This author contended that from the perspective of vulnerabilities, programs were the same as other data. While the consequences of unauthorized modification of a program might be very different from the consequences of the modification of other data, we had agreed not to treat consequences.

The majority believed that more rigorous controls were indicated for programs than for other data. This author contended that that indication stemmed from the consequences and not from the vulnerabilities. I further contended that the distinction drawn between programs and other data had not resulted in more rigorous controls over programs; that indeed other data is generally much better controlled than programs are. I contended that treating programs and other data together would

result in an improvement in the control over programs, at least to the level afforded other data. In spite of these persuasive arguments, there was no consensus. If the auditor feels that the vulnerabilities of programs are different than those of other data, then different controls may be indicated.

3.1.2 Vulnerabilities of Terminals: The group concluded that terminals were vulnerable to damage, theft and unauthorized use. Theft can be viewed as whole or partial theft, and damage as reparable or irreparable. The vulnerability of the terminal to conversion or damage is related to the terminal characteristics discussed in paragraph 2.4. Its susceptibility to unauthorized use is primarily a function of the environment in which it is placed. However, it may be marginally related to characteristics or features of the terminal such as mag-stripe card readers or locks, and administrative controls in place at the terminal.

3.1.3 Vulnerabilities of Media: Media were seen as being vulnerable to the same hazards as terminals, i.e., damage, theft or unauthorized use. However, media may be significantly more vulnerable to these things than terminals and the consequences may be significantly more severe. Media are much more readily stolen or damaged. The consequences may include disclosure or destruction of the data. The media types included in figure 1 were considered against ten different characteristics. While we believe figure 1 to be reasonably complete as it relates to today's technology and to the near term future, the auditor is cautioned that fully half of the items on this list have been introduced within the last five years. However, we believe most future media can be readily described and evaluated in terms of the following characteristics.

3.1.3.1 Readability: Readability is simply the ease with which information stored on the media can be retrieved. It can be measured in terms of the cost or the availability of the technology required to retrieve the data. By this measure paper and CRT would be considered more readable than tapes or disks. Microfilm, while less readable than paper, may be considered more readable than tape or disk. Mass storage cartridges might be considered less readable than tape, if only because mass storage devices are more expensive and less numerous than tape drives. Readability can also be measured in terms of the distance at which the media can be read. Under this measure CRTs would be considered more readable than paper for not only does the CRT emit light rays but the electron beam which draws the characters on the screen carries information which can be detected at a distance. In general, the vulnerability of data increases with readability and decreases with distance.

3.1.3.2 Density: Density is a measure of the quantity of information stored on the medium as a function of its total volume. For example, paper is more dense than CRT and less dense than microfilm. A mass storage cartridge might be more dense than tape and less dense than microfilm. In general, the vulnerability of media will go up with its density.

3.1.3.3 Portability: Portability is simply a measure of how readily or easily the media can be carried. For example, a disk pack might be considered less portable than a mass storage cartridge and more portable than a terminal. In general, the vulnerability of the medium to theft will increase with its portability.

RIGID DISK
 FLOPPY DISK (ETTES)
 FIXED-HEAD ASSEMBLY
 MAG-TAPE (BY REEL SIZE)
 MAG-CARDS
 TAPE CARTRIDGES
 TAPE CASSETTES
 PAPER (BY SIZE)
 PAPER TAPE
 CARDS
 FICHE
 ROLL MICRO-FILM
 REMOVABLE NON-VOLATILE
 (BUBBLE) MEMORY
 "CHIPS"
 TERMINAL SCREEN OR KEYBOARD
 OTHER

MEDIA TYPES

Figure 1

3.1.3.4 Size: Related to the density and portability of the medium is its size. In general, the smaller the medium the more vulnerable it is to theft, since the smaller it is the more readily it may be removed without detection. Thus, a tape reel might be more vulnerable than a disk pack and less so than a floppy diskette or a removable memory chip. While density, portability and size may be considered independently, they are closely related and may also be usefully considered together.

3.1.3.5 Permanence: Permanence, as used here, refers to the capacity of the medium to resist the modification or erasure of its data. Thus, paper would be considered more permanent than magnetic tape. On the other hand, tape might be considered more permanent than some "chips" that require the continued presence of a magnetic field or voltage in order to retain data. In general, the vulnerability of the data itself to modification or destruction goes down with permanence. However, the vulnerability of the medium to theft goes up with permanence.

3.1.3.6 Value: The vulnerability of the medium to theft increases with its intrinsic value. Thus, one might be more likely to steal floppy disks or mag cards than paper or punched cards.

3.1.3.7 Integrity: Integrity is used here in the very narrow sense of indivisible. In this sense, a tape reel has less integrity than a floppy disk and more than a card deck. In general, the vulnerability of both the data and the medium increases as integrity decreases. For example, a very large card deck could be stolen one card at a time. Since information is almost invariably stored in the context of the data, the integrity of the data depend at least in part upon the integrity of the medium. For example, in a series of records, the absence of record C may be implied by the fact that record D follows immediately after record B. This implication can be more reliably stored on tape than in a card deck.

3.1.3.8 Authenticity: Authenticity is used here to describe the ability of the medium to resist substitution or counterfeiting. Thus, a tape labeled both on the reel and on the header might have more authenticity than a card deck, but less than a disk pack with an engraved serial number. In general, the vulnerability increases with a decrease in authenticity.

3.1.3.9 Flamability: Flamability is one of the key measures of the susceptibility or vulnerability of a medium to physical damage. For example, magnetic tape would be less flammable than paper and more flammable than a magnetic disk.

3.1.3.10 Frangibility: Frangibility is the susceptibility of the medium to breakage. For example, a magnetic disk may be more frangible than a magnetic tape and less so than a memory module.

3.1.4 Vulnerability of Services: The working group concluded that one of the vulnerabilities to the system that results from remote processing is the conversion of service or capacity from the use of the owners to the use of another person. Most often this other person will be an employee, but under some circumstances he may be a vendor or an outsider.

The vulnerability of the system to conversion of service is a function of the generality and marketability of the service. Thus, a personal computing system could readily be converted to the use of one of its users. A transaction-driven business system, on the other hand, while vulnerable at the data and application level, is significantly less vulnerable at the service level.

3.2 Hazards

Having identified the system vulnerabilities as a step toward identifying the appropriate controls, the working group felt that it would also be useful to examine the hazards. They were divided into natural and man-made.

3.2.1 Natural Hazards: Natural hazards are all hazards except Man. They include such things as fire, wind, earthquake, rising and falling water, and lightning.

3.2.2 Man-made Hazards: Man-made hazards can usefully be separated into accidental and intentional.

3.2.2.1 Accidental Hazards: Accidental hazards include all errors and omissions. For example, data may be incorrectly recorded or transcribed or the recording of data may be completely omitted; a terminal may be dropped or damaged by something dropped upon it; media may be lost misplaced or mislabeled; services such as connect time can be accidentally wasted by the omission of a dial disconnect.

3.2.2.2 Intentional Man-made Hazards: Intentional man-made hazards can usefully be viewed by type and by method.

o Types of Intentional Man-made Events: Intentional man-made events may include mischief, vandalism, riots, wars, theft, fraud, embezzlement, and other types of conversion.

o Methods of Attack: In developing its recommendations on controls, the group considered eight specific methods by which information systems might be intentionally attacked.

1. Browsing: As its name implies, browsing is scanning available data in an attempt to identify and exploit sensitive data. Examples might include examining media stored in the remote site or using the remote terminal or device to examine information stored in the host system using normal access facilities.

2. Eavesdropping: Eavesdropping is a special case of browsing characterized by the fact that the attacker is outside the controlled environment. Examples might include observing a CRT from a distance using a telescope or collecting acoustic emanations from a typewriter terminal by the use of a parabolic microphone, i.e., non-normal access.

3. Exhaustive Attack: An exhaustive attack is a means for determining secret or confidential data by trying all of the possibilities and testing for the correct possibility. For example, one can always discover a correct password providing that one is able to try enough different possible passwords.

4. Spoofing/Posing: Spoofing or posing is an attack in which a person or process pretends to be a more privileged person or process. For example, if a person is able to determine an ID and password of a system user, he may then pretend to the system to be that person.

5. Trojan Horse: The Trojan Horse is an attack in which a hostile entity is concealed inside an innocent one for the purpose of getting it through a protective perimeter. For example, a fraudulent transaction could be entered into the system by concealing it among a large number of legitimate transactions.

6. Trap Door: A trap door attack is a special case of a Trojan Horse attack used in the face of compartmentation defenses or separation of duties. It provides a secret door between the compartments known only to the attacker. For example, a programmer may insert a trap door that he can use in order to appear as a legitimate user.

7. Time Bomb: A time bomb is another special case of a Trojan Horse attack in which the hostile process is triggered by an event in time which need not be under the control of the attacker. For example, a programmer might insert code for his own purposes which is triggered by the system's time of day clock.

8. Asynchronous Attack: Asynchronous attacks are those which attempt to exploit the time between a defensive action and the attack itself in order to nullify the effect of the defensive action. For example, a person might attempt to gain use of a terminal after a legitimate user had logged on, but before he had logged off or been timed out. In this way he might avoid the checks for a legitimate user that take place at log on time.

4. CONTROLS

The working group identified and articulated the controls which it believes to be indicated and effective against these vulnerabilities and hazards. We have attempted to present these controls in the manner which is most useful to an auditor reviewing security in a location with remote terminals or peripherals. It is our hope that this method of presentation will also be useful to systems designers and managers.

4.1 Control Principles

In identifying the specific controls that should be considered for use in an environment including remote terminals or peripherals, the working group was guided by the following principles.

4.1.1 Separation of Duties: In general, risk can be reduced and security can be improved by involving multiple people in sensitive duties. Management's ability to separate duties may be limited both by scale and by a desire to maintain other benefits. However, the following tests should still be met.

4.1.2 Restrict Access: Access to sensitive resources such as terminals, media and data should be restricted. In general, access should be restricted such that a person has access to only those sensitive resources to which he must have access in order to be able to carry out his assigned duties.

4.1.3 Independent Authorization: Sensitive activity and transactions should be subject to independent (management) review, approval and authorization. Examples might include authorizing a user access to a system and explicit authorization to enter a particular class or type of transaction; execution by management of a transaction specifically required to approve a transaction or a group of transactions previously entered by nonmanagement personnel; review by management of a subset of transactions selected by the system and reported to management.

4.1.4 Individual Accountability: It should be possible to fix accountability for every significant event to the level of a single individual. Likewise, it should be possible to relieve individuals of accountability for all acts which they did not commit.

4.1.5 Test of Concealment: Duties should be assigned, access restricted, activity approved, and accountability fixed such that no single individual can both fail in his duties, accidentally or intentionally, and conceal that fact.

4.1.6 Test of Sensitive Combinations: Duty should be assigned and access restricted such that no one has access to a sensitive combination of resources. Sensitive combinations include access to a resource and to the control records for that resource, the ability to originate a transaction and approve the same transaction, the ability to maintain a record and process transactions against the record, and the ability to process a transaction and change the rules under which the transaction is to be processed.

4.2 Control Measures

The working group recommends the following control measures. While they are based upon the control principles in paragraph 4.1, and designed to address the vulnerabilities and hazards articulated in paragraph 3, no attempt has been made to relate or associate the control measures with specific vulnerabilities or hazards. Instead the measures are presented in categories with similar measures and the categories are presented in the order in which they may be convenient for the auditor to test for them. Therefore, the auditor is cautioned that not all controls will be indicated, necessary or required in all environments.

4.2.1 Explicit Assignment of Responsibility: The auditor should expect to find that duties and responsibilities have been explicitly assigned. More specifically, he should expect to find:

4.2.1.1 Assignment of Security Responsibility: Responsibility for the custody and protection of the terminal, media and data should be explicitly assigned.

4.2.1.2 Access Rules: Rules as to who may access terminals, media and data, who may authorize access to terminals, media and data, and how such authorizations will be recorded should have been explicitly defined.

4.2.1.3 Control Principles: Assignment of responsibilities outlined in paragraphs 4.2.1.1 and 4.2.1.2 should be consistent with the control principles articulated in paragraph 4.1.

4.2.1.4 Environmental Tests: Assignment of responsibilities called for in paragraph 4.2.1.1 and 4.2.1.2 should give proper weight to the environmental parameters articulated in paragraph 2.

4.2.2 Physical and Environmental Controls: Proper consideration should be given to the vulnerabilities and hazards and control principles when selecting the physical environment for the terminal and its media. When considering protection from natural hazards, first consideration should go to the safety of people. Normally, that environment which is safe for people will also be safe for the terminal and media. However, concern for manmade hazards will indicate additional requirements. The physical environment should facilitate the enforcement of the rules of access to the terminal and the media. This suggests that it is desirable to install the terminal in a small room with low occupancy. The environment should facilitate the timely detection of variances or losses. This suggests that the terminal should not be installed in a room that is frequently unsupervised. And finally, the physical environment should facilitate the fixing of accountability. This requirement suggests that it is desirable to have the terminal in the same environment as its normal users, and not in the same environment as anyone who is not among its normal users.

4.2.3 Access Control: A combination of physical, administrative system and application controls must be in place to consistently enforce the rules (see paragraph 4.2.1) as to who may have access to terminals, media and data. Environmental controls for the terminal were treated in paragraph 4.2.2. Where indicated by this environment additional controls for control of access to the terminal such as key locks or management supervision should be considered. In particularly hostile environments it may be necessary to physically secure the terminal to the desk or building. Alarms which are triggered when an attempt to move the terminal is made may also be useful. Cover locks may be indicated to prevent the removal of media or components from the terminal. Where large quantities of media are used with the terminal, provisions should be made for its safe keeping. Most often these provisions will involve lockable cabinets. In general, the control of access to data in the host system will be provided by application and/or system controls. The discussion of such controls is treated in a different report. However, the auditor is cautioned that the selection and adequacy of local controls may be balanced against and must be appropriate to the application and system controls.

4.2.4 Audit Trail: Records must be kept such that the auditor can establish that controls are in place and that they are uniformly and consistently applied. More specifically, records must be kept such that:

- o The assignment of responsibility and establishment of access rules consistent with Control Principles can be demonstrated. Normally, it will be possible to make this demonstration from documents kept for other purposes. These documents may include policy statements, standards and guidelines, procedures, program specifications, job descriptions and performance plans.
- o Consistent enforcement of access rules can be demonstrated. This may involve the keeping of records specifically for this purpose. Such records might include a log of all attempted accesses distinguishing between those allowed and those disallowed.
- o Accountability for service consumption and resource (data, media, terminal, etc.) use can be established. Normally this portion of the audit trail would be composed of system accounting records.
- o The presence or authorized absence or use of media can be demonstrated. This portion of the audit trail will usually be composed of media inventory control records such as the library management system.
- o That required approvals and authorizations were given. This portion of the audit trail will consist of such things as transaction source documents, program specifications and change orders along with management signatures affixed thereto.

All of the items that make up the audit trail should contain reference to their environment, i.e., who, what, where, when and how. (Since the environment of a part of the audit trail includes all other parts, these requirements are often met by cross reference.)

4.2.5 Contingency Plans: Adequate emergency, backup and recovery plans should be developed.

4.2.5.1 Emergency Plans: Management should have plans in place for identifying and containing the damage that might be associated with catastrophic or man-made events. These plans should deal with fire and other natural disasters and intrusions or similar man-made events. They should involve alarms and notifications, shelter or evacuation as indicated, and materials and procedures for damage control.

4.2.5.2 Backup Plans: In the event of damage so severe as to deny the use of the system for an extended period, management must have planned alternative methods to satisfy the requirements normally met by the system.

More specifically, an acceptable alternative method of accomplishing the applications normally performed by the host system must have been identified. For nondiscretionary systems such as business transaction systems this may involve the substitution of manual procedures, and dial voice communications. For discretionary systems such as application development or personal computing the plan may involve deferral or the use of alternative systems.

4.2.5.3 Recovery Plans: Management should also have plans for the permanent restoration of access to and use of the host system service. Alternative sources of replacement terminals or peripherals should be identified in advance. Sources should include multi-plant vendors or multiple vendors. Where compatible terminals might not be readily available, application changes which would permit substitution of non-compatible devices should have been planned. Alternative sources of required media must be identified. Such sources may include secondary inventories within the same organization or multiple vendors. Provisions must exist for recovering or reconstructing data. Such provisions may include keeping natural copies of the data in both the host and remote locations. In some cases, recovery and reconstruction of data lost at the host may involve reprocessing of original source documents from the remote location. Residual risk must be identified, quantified, accepted as a business risk, or assigned to insurers.

4.2.6 Test and Reconciliation: Procedures must be in place to compare and reconcile on a timely basis the behavior, use and content of the system to expectation. Such expectations may be imposed external to the organization, by management or by the system. External expectations will include such things as accepted practices, laws and regulations and express or implied contractual obligations. Management expectations will include such things as policy, standards, guidelines, mission or duty assignments, interdepartmental agreements and procedures implementing any of the above. System imposed expectations may include the specifications for the system, the terminal, the communication protocol, application system rules, hardware specifications, specifications of vendor-supplied software and security or access rules.

4.2.6.1 Reconciliation of Data: The auditor should be able to satisfy himself that data and programs are being reconciled to expectation. The expectations will include program specifications, transaction authorizations, and the external environment. Reconciliation techniques may include reports and confirmations. The auditor should satisfy himself that such reports and confirmations are being reconciled, that variances are being identified and that corrective action is being taken.

4.2.6.2 Reconciliation of Use: The auditor should satisfy himself that resource use is being reconciled to expectation including access rules and authorizations. Where the value of the resource is significant, it should be billed or charged to the user's manager with copies to the user and to the owner of the resource. The user's manager should reconcile such use to his own expectation including the user's job assignment. Variance between actual use and that which might be expected from the user's job assignment may represent unauthorized use on the part of the user. The user should reconcile the billing to his actual use of the system. Variances between actual use and billed use may represent unauthorized use of the system in the user's name. The owner of the resource should reconcile such use to his expectation including previous use, the plan and the budget. Variances may represent unauthorized use coupled with a failure to reconcile on the part of users and managers.

4.2.6.3 Reconciliation of Security Variances: The auditor should satisfy himself that variances from security rules are being recognized and appropriate corrective action taken. Such variances may represent user errors caused by poor system design or inadequate user training. They may further represent a casual or systematic attempt to penetrate the system. Management failure to take prompt corrective action may result in waste and may encourage further attempts to breach the controls of the system.

4.2.6.4 Reconciliation of Property: The auditor may wish to satisfy himself that management systematically reconciles the controls over physical resources such as terminals and the media. In the absence of reports of such reconciliations the auditor may wish to make a physical inventory of his own. Failure to reconcile such controls may encourage casual or systematic conversion on the part of employees.

4.2.6.5 Tests of Contingency Plans: Finally, the auditor will wish to examine evidence that contingency plans are in place and are being tested in a systematic way on a regular frequency. Such evidence may include records and reports of drills and tests. Failure to conduct such drills and tests reduces the probability that the plans will work as written.

PART VII: SESSION 5

COMMUNICATION COMPONENTS

Chairperson: Jerry FitzGerald
Jerry FitzGerald & Associates

Participants:

Dennis K. Branstad
National Bureau of Standards

Milton Lieberman
Spectron Inc.

P. J. Corum
Toronto Dominion Bank

Aileen MacGahan
Chase Manhattan Bank

Steve Kent
Massachusetts Inst. of Tech.

David A. Rubin
Peat Marwick Mitchell & Co.



From left to right: David A. Rubin, Milton Lieberman, P. J. Corum, Jerry FitzGerald, Dennis K. Branstad, Steve Kent, Aileen MacGahan.

Note: Titles and addresses of attendees can be found in Appendix B.

EDITOR'S NOTES

JERRY FITZGERALD

Dr. Jerry FitzGerald is the principal in Jerry FitzGerald and Associates, a management consulting firm located in Redwood City, California. He has extensive experience in data communications, data processing security, and EDP auditing.

As a consultant, he has been active in numerous EDP audit reviews, management development/reviews of the internal EDP audit function, EDP security assurance reviews, and data communications/teleprocessing projects (especially those involved with on-line distributed networks). In addition to consulting in EDP auditing, data processing security, and data communications, Dr. FitzGerald has developed state-of-the-art training seminars in these three areas.

Prior to establishing his own firm, Dr. FitzGerald was a Senior Management Consultant with SRI International (formerly Stanford Research Institute), an associate professor of data processing/accounting in the California State University and Colleges System, and has held various other senior positions within private industry and governmental organizations.

Dr. FitzGerald's educational background includes a Ph.D. in business administration, an M.B.A., and a Bachelor's Degree in industrial engineering. He has written extensively on data communications, EDP auditing, and data processing security. His current books are Internal Controls for Computerized Systems, Fundamentals of Data Communications, and Fundamentals of Systems Analysis.

THE CHARGE GIVEN TO THE GROUP

All modes of data transmission were to be considered. Specific vulnerabilities were to be identified along with appropriate safeguards, e.g., interception of microwave transmissions, with encryption serving as the countering control. [See PART I, Section 2 for the complete charge given to this group.]

The report that follows is the consensus view of this session.

AUDIT AND CONTROL OF COMMUNICATION COMPONENTS

Jerry FitzGerald, Chairman

and (alphabetically listed):

Dennis Branstad	"P" "J" Corum
Stephen Kent	Milton Lieberman
Aileen MacGahan	David Rubin

INTRODUCTION

This paper is a follow-on to the first National Bureau of Standards (NBS) invitational workshop on audit and evaluation of computer security. The earlier paper was published in NBS Special Publication 500-19 (Part X).

In this second paper, the committee presents a set of guidelines that can be used when conducting a review of administrative and technical controls pertaining to a multiple user teleprocessing environment. The committee intends that this paper form the basis upon which auditors or security experts might review the degree of adequacy contained in the controls within a teleprocessing network.

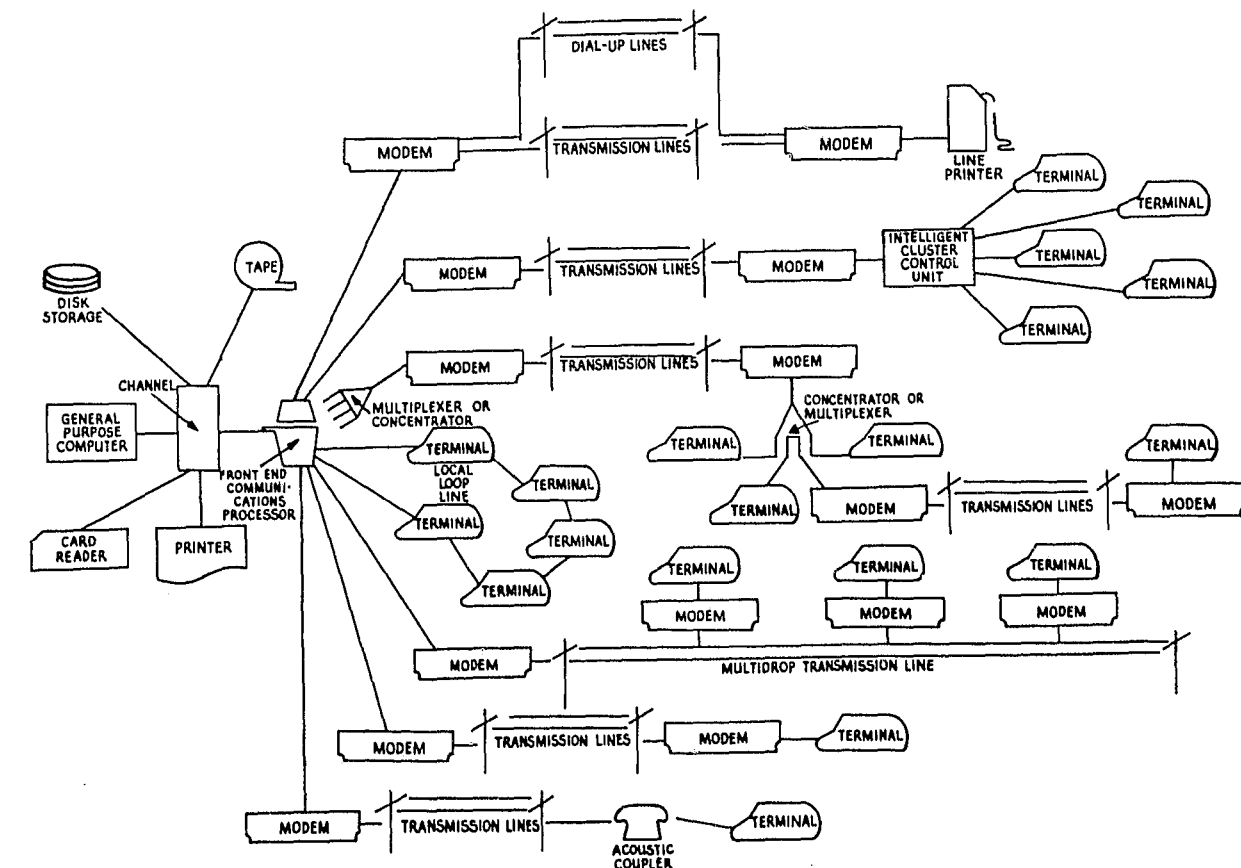


FIGURE 1: NETWORK CONFIGURATIONS

In order to better understand what is meant by a teleprocessing environment, the preceding figure (Figure I) was developed to show examples of the alternative teleprocessing network configurations that might be available. These networks are among those that might be faced when conducting a security review in today's teleprocessing environment. It should be noted that there might be combinations of networks, where for example a multi-drop configuration might have a local loop at each of the drops. Also, where this figure depicts "transmission lines" the audit and control expert reviewing the network might find various transmission media, such as satellite circuits, microwave transmission, fiberoptics, or copper wire pairs.

DEFINITION OF THE COMMUNICATION COMPONENT SECURITY AUDIT

For the purpose of this paper a computer security audit is defined as an independent evaluation of the controls employed to ensure the accuracy and reliability of the data maintained on or generated by a teleprocessing network, the appropriate protection of the organization's information assets (including hardware, software, and data) from all significant anticipated threats or hazards, and the operational reliability and performance assurance of all components of the automated data processing system.

With regard to the communication component, all modes of data transmission and associated equipment should be considered. Specific vulnerabilities should be identified along with appropriate safeguards, e.g., interception of microwave transmissions, with encryption serving as the countering control.

THE CONTROL MATRIX

This paper presents a matrix that relates the various vulnerabilities to the specific controls that might be available to mitigate them (see Figure II, The Control Matrix). The vulnerabilities are listed across the top of the matrix and are defined in a later section of this paper. The controls are listed down the left vertical axis of the matrix and are also defined in a later section of this paper. Within the cells of the matrix there is either an X or an O whenever the control is an appropriate countermeasure to a specific vulnerability. An X indicates a primary control that can be used to mitigate the specific vulnerability; an O indicates a secondary control that might be useful in mitigating the specific vulnerability. To apply the matrix, first identify the vulnerability that may be present in your teleprocessing network. Next, proceed down the column of the specific vulnerability and identify whether the controls in the left vertical column are applicable.

The control matrix can be used in two other ways to assist the auditor. The first is to determine the exposures that will be faced by the organization whenever one of the vulnerabilities does, in fact, occur. These exposures are listed at the bottom of the matrix, below each vulnerability column. For example, if the vulnerability "Message Lost" occurred, then the organization would be subjected to exposures A, E, F, and G. These exposures are defined in Table I.

The second use to which the matrix can be put is to specifically identify the various components of the network where the controls might be most effectively located. To do this, the auditor would choose a specific control such as "Sequence Number Checking" and follow across that row to the right-hand side of the matrix, where there are some numbers, such as 9, 10, 17. These numbers indicate those specific components of a data communication network where the controls might be located. These 17 components are defined at the end of this report.

INTERRELATIONS OF SECURITY CONTROLS

The auditor should recognize that the security controls shown in the matrix have complex interrelations in solving certain security problems. There are no linear equations that show how these controls add to or subtract from one another. The security controls required in a worst case analysis of an intentional assault on a communication system constitute a highly structured set of interrelationships.

FIGURE II: THE CONTROL MATRIX

CONTROLS	VULNERABILITIES									COMPONENTS WHERE CONTROLS ARE LOCATED
	MESSAGE LOST	MISROUTING	MESSAGE ALTERATION	DISRUPTION	DISASTER	DISCLOSURE	MESSAGE INSERTION	THEFT (Physical)	DUPLICATE MESSAGE	
SEQUENCE NUMBER CHECKING	X	O					X		X	9,10,17
SENDING AND RECEIVING IDENTIFICATION	O	X				O				9,10,17
TRANSACTION JOURNAL	X	O	X	X	X		X		X	9,10,11,17
POSITIVE ACKNOWLEDGMENT	X	X	X	X			X		X	9,10,11,17
TIME AND DATE STAMP	X								X	9,10,11,17
PERIODIC MESSAGE RECONCILIATION	X		X	O			X		X	10,17
CHECK SUM ON MESSAGE ADDRESS		X								9,10,11,17
ERROR DETECTION CODE			X	O						9,10,11,12,17
ERROR CORRECTION CODE			X	O						9,10,11,12,17
KEY REDUNDANCY CODE			X	O						9,10,11,12,17
ECHOPLEXING	X	O	X	O			O		O	10
ERROR LOGGING	O	O	X	X					O	9,10
BACKUP EQUIPMENT AND FACILITIES				X	X					1,7,8,9,10,11,17
PHYSICAL SECURITY	O	O	O	X	X	X	O	X	O	1-17
RECOVERY PROCEDURES	O		O	X	X				O	1-17
COMMUNICATION POLICY	O	O	O	O	O	O	O	O	O	1-17
LIFE SUPPORT SYSTEM				X	X					4,8,9,10,11
DEVICE DISCONNECTION DETECTION	O							X		7,9,10,11
BUILT-IN DEVICE ADDRESS							X	O		9,10,11,17
ENCRYPTION	X	X	X			X	X		X	7,8,9,10,11,17
UNLISTED PHONE NUMBER (Dial-Up)						X	X			9,10,11
LOW ERROR RATE FACILITIES	X		X	X						1,2,3,7
SOFTWARE CONTROLS AND TESTING	X	X	X	X		X	X		X	9,10,11,17
DOCUMENTATION	O	O	O	O	O	O	O	O	O	1-16
EMANATION CONTROL						X				1-4,6-11,13-17
TRAINING AND EDUCATION	O	O	O	O	O	O	O	O	O	1-17
EXPOSURES (See Table I)	A,E, F,G	A,D, E,F, I	A,D, E,G, H	C-I	C-I	F,G, I	B,D, E,G, H	C,E, G,H	D,E, G,I	

TABLE I: EXPOSURES

- A. Erroneous Record Keeping
- B. Unacceptable Accounting
- C. Business Interruption
- D. Erroneous Management Decisions
- E. Fraud
- F. Statutory Sanctions
- G. Excessive Costs/Deficient Revenues
- H. Loss or Destruction of Assets
- I. Competitive Disadvantage

The above items A through I represent the various exposures that the organization faces whenever some sort of a vulnerability (threat or concern) takes place. In other words, the result of a threat might be one of these exposures.

For example, encryption is a valuable security control in a communication system. It is not, however, a complete solution in and of itself. The security objectives of a communication system can only be satisfied when encryption is used in conjunction with several other controls. In particular, sequence numbers must be used to detect attempts to add, delete, or replay messages by a technically competent penetrator. A cryptographic error detection code must be used to detect alteration of messages. Encryption key management must be performed to ensure authentication of communicating devices.

In addition, message reconciliation must be performed during and at the end of every session to ensure that all messages transmitted have been received. Emanation controls prevent the loss of encryption keys and plaintext messages through undesirable electronic phenomena.

These constitute the necessary set of nondiscretionary controls required for secure communication. In addition, certain discretionary, human-oriented controls are required to support the encryption system. Physical security must prevent theft or unauthorized use of a device containing a valid encryption key. Maintenance and testing must ensure the correct operation of the controls. Documentation must explain how the controls must be used. Finally, the user must be educated and trained in the use of these controls.

DEFINITION OF THE VULNERABILITIES

The following list defines the vulnerabilities that are listed across the top of the control matrix. These vulnerabilities could be interpreted as the concerns or threats to which a data communication network might be subjected.

- Message Lost: Refers to a message that never reaches its intended destination.
- Misrouting: Is said to occur in a message-switching network when a message intended for a destination, e.g. Node A is sent to another destination, Node B.
- Message Alteration: Refers to unauthorized (accidental or intentional) modification of an authentic message.
- Disruption: A temporary or intermittent service outage affecting one or more of the network components which may result in one or more of the following consequences: denial of service, misrouting, message alteration, messages lost, duplicate message, etc.

- Disaster: An interruption resulting in denial of service for an extended period of time as the result of an accident, natural catastrophe, or sabotage. The distinction between a disaster and a disruption is based upon the length of service outage and upon the permanence of the damage to the affected components.
- Disclosure (Privacy): Unauthorized access to any data is disclosure. If the data is personally identifiable to an individual or legal person, then the unauthorized disclosure is a privacy violation.
- Message Insertion: The addition of an extraneous unauthorized message at any component in the network. This vulnerability is never accidental and does not include duplicate messages.
- Theft (Physical): Physical theft refers to unauthorized removal of any hardware component.
- Duplicate Message: The insertion or processing of multiple copies of an otherwise authorized message. This can occur accidentally or intentionally.

DEFINITIONS OF THE CONTROLS

The following list defines each of the controls listed down the left vertical axis of the control matrix.

- Sequence Number Checking: A method where all messages contain an integral sequence number for each level of the communication system. Verification techniques must detect duplicate and missing numbers, reject duplicates, and report missing messages.
- Sending and Receiving Identification: A method where sufficient information is contained in the message to uniquely identify both the sender and the receiver of a message.
- Transaction Journal: A method of capturing sufficient system and message level data to establish an adequate audit trail or to have an actual copy of each and every transaction transmitted in the network.
- Positive Acknowledgment: A method where the receipt of each message is positively confirmed back to the sender.
- Time and Date Stamp: An automatic procedure whereby each message contains time and date information for each major processing node.
- Periodic Message Reconciliation: System facilities to verify completeness of processing by periodically providing summary information to reconcile number of messages, dollar values, control totals, etc., both sent and received.
- Check Sum on Message Address: A procedure that verifies the message address using hashing or other summing type of totals.
- Error Detection Code: A method of inserting redundant information for purposes of detecting any changed bit patterns.
- Error Correction Code: A method of inserting extra (redundant) bits of information to permit detection and correction of errors at the receiving equipment without retransmission of the original message.
- Key Redundancy Code: The insertion of duplicate information in key fields of the message stream (such as dollar amounts, description identifiers, quantities, etc.) which can be compared at the receiving equipment for correctness.
- Echoplexing: A verification procedure by which each character received by the receiving station equipment is transmitted back to the originating equipment.

- Error Logging: A software program that records error messages, by line, terminal, and also type and frequency. This recording is to measure the degree of reliability and performance of the communication system. Statistical analysis and management reports are required for evaluation and corrective action to minimize error rates.
- Backup Equipment and Facilities: Duplicate or alternate equipment (power, air conditioning, etc.), software, and procedures to be invoked whenever a major outage occurs with the primary system. Also a physical facility located away from the primary site and capable of supporting the original primary site telecommunication function at an acceptable operational level.
- Physical Security: The ability to have proper physical security over the data communication facilities, software, and all other aspects of the teleprocessing network. This includes restrictive access controls over personnel, adequate fire protection, backup electrical equipment, and any other aspects of physical security with regard to maintaining the integrity of the data communication network.
- Recovery Procedures: A set of written procedures that clearly defines responsibilities and procedures for operational programming and supervisory personnel to allow for the orderly recovery of the system to operational status or to recover from excessive error rates.
- Communication Policy: A statement of agency or corporate policy regarding design, use, and maintenance of communication components including security objectives and penalties for not achieving these objectives.
- Life Support System: Equipment, techniques, and procedures that will eliminate or minimize damages caused by disasters, occurrences such as fire, power failures, flood, environmental changes, etc.
- Device Disconnection Detection: The use of electrical control signals or other mechanisms to detect physical disconnection of communication system components.
- Built-in Device Address: The imbedding of a device address or identifier via hardware or software mechanisms in communication system components.
- Encryption: The transformation of data (cleartext) to an unintelligible form (ciphertext) through the use of an algorithm under the control of a key such as the federal Data Encryption Standard (DES) (FIPS Pub. 46).
- Unlisted Phone Number (Dial-Up): The acquisition and use of unlisted telephone numbers for the communication system component that can be accessed via dial-up lines.
- Low Error Rate Facilities: The selection and use of data transmission facilities with characteristically low error rates such as conditioned lines or digital transmission lines.
- Software Controls and Testing: The procedures employed in development, installation, and maintenance of software in communication system components to insure the correctness, integrity, and availability of the software.
- Documentation: The generation, revision, and maintenance of manuals dealing with appropriate design, maintenance, and operational aspects of the communication system.
- Emanation Control: The use of shielding and associated techniques to suppress electromagnetic, acoustic, and radio frequency emanations from communication system components.
- Training and Education: The development, presentation, and periodic review of educational materials dealing with correct operation and maintenance of the communication system.

GENERAL DEFINITIONS OF COMPONENTS

The following list of items enumerates and defines the components of a data communication network. In some cases the item listed may be a characteristic of data transmission rather than an actual component.

1. Circuits: A circuit can be a single communication facility or a combination of different types of communication facilities such as:
 - Satellite: A facility that uses ultra-high frequency signaling relayed through a device orbiting the earth.
 - Microwave: A facility that uses high frequency signaling which passes through terrestrial relay points.
 - Fiber optics: A facility that transmits signals through the use of optical media utilizing a fiberglass-like cable.
 - Wire: A facility that transmits through a metallic conductor. This facility may utilize long-distance copper wire pairs, coaxial cable, or the copper wire local loop between a user premises and the telephone company's switching office.
2. Analog Transmission: Transmission of a continuously variable signal which has an almost infinite number of states (an example of an analog signal is a sine wave).
3. Digital Transmission: Transmission of a discretely variable signal such as discrete voltage levels (an example is signaling which is composed of either a positive or a negative voltage).
4. Carrier Switch/Facility: A communication facility supplied by a commercial vendor of telecommunication services that provides for the interconnection of transmission devices (an example would be the telephone company's switching office or the Telnet Packet switches).
5. Configurations: These are the methods of connecting communication devices. There are many examples of communication configurations, some of which were shown in Figure I. Examples of these configurations might be as follows:
 - Dedicated/private leased lines. These circuits are always available to the customer for transmission and generally are used with on-line real-time systems.
 - Dial/switched circuits. A circuit connection which is established by dialing a telephone or establishing a physical or logical connection before data can be transmitted.
 - Point to point circuits. This method provides a communication path between two points. It can be a dial-up or a dedicated circuit.
 - Multidrop circuits. This method allows for the sharing of a communication facility. It is similar to a party line telephone call because several input/output terminals share the same line. Only one terminal can be transmitting on the line at a time.
 - Local cable. This method of connecting communication devices consists of a privately owned cable or wire interconnecting many terminals with the computer system.
6. Packet Switching (Value Added Networks -- VAN) System: A type of data communication technique that allows for messages to be divided or segmented into packets and routed dynamically through a network to the final destination point.
7. Interface Unit: The device that connects a data transmitting (terminal) or receiving unit to the transmission facility. An example of this would be a modem, a digital service unit, or a device that converts voltage signaling to light signaling.

8. Multiplexer: A device that combines several independent data streams into one data stream at a higher signaling speed for transmission to a similar device that separates the high-speed signal into the original independent data streams. Note: Some of the multiplexers are software-driven and are similar to concentrators; however, most of them are non-intelligent hard-wired devices.
9. Concentrator: A programmable device that will perform the same function as a multiplexer with added functions such as data storage (buffering), message error checking, data flow control, polling, etc.
10. Front-End Communication Processor: A programmable device that interfaces a communication network to a host computer. Some of the functions that can be performed by a "front-end" are polling, code and speed conversion, error detection and correction, store and forward functions, format checking, data flow control, network statistics gathering, message authentication, communication routing and control, and the like.
11. Message Switch: A privately owned programmable device that accepts messages from many users, stores them, and at some time after receiving them transmits them to their intended destination. This device generally receives messages at slow speeds over dial-up lines.
12. Protocols: Software or hardware rules that facilitate the transmission between devices. Some protocols provide for error control.
13. Test Equipment (technical control facility): A combination of equipment that facilitates the physical monitoring, diagnostics, and restoration of communication systems should they fail. They can contain circuit patching, spare equipment, alternate switches, and might involve message text monitoring or quantitative measuring equipment.
14. Audio Response Unit: A unit that accepts analog, audio voice, or digital signals and converts them to digital computer signaling or can also convert digital signals from a computer into human understandable voice signals.
15. Auto Answering: A device that automatically answers a telephone and establishes a connection between data communication devices.
16. Auto Dialing Unit: A device that accepts computer signals and automatically dials the telephone number of a remote communication device.
17. Terminals: An input/output device that is used to enter messages into the system and/or receive messages from the system.

PART VIII: SESSION 6

PROCESSORS, OPERATING SYSTEMS, AND NEARBY PERIPHERALS

Chairperson: Theodore M. P. Lee
Sperry UNIVAC

Participants:

Peter Neumann
Stanford Research Institute

Peter Tasker
MITRE Corporation

Gerald J. Popek
Univ. of California, LA

Stephen T. Walker
CCC&I, Dept. of Defense

James E. Rife, Recorder
U.S. General Accounting Office

Clark Weissman
System Development Corporation



From left to right: Peter Tasker, James E. Rife, Peter Neumann, Clark Weissman, Gerald J. Popek, Theodore M. P. Lee, (Stephen T. Walker absent).

Note: Titles and addresses of attendees can be found in Appendix B.

EDITOR'S NOTES

THEODORE M. P. LEE

Dr. Theodore M. P. Lee is Manager, Systems Security, for Sperry Univac's major systems development organization, where he is the focal point for the security aspects of all present and future products, hardware and software. Just prior to assuming his present position in late 1978 he was a staff consultant in Sperry Univac's Product Strategy and Requirements organization, working as part of a large R&D project for potential future products, where he had major roles in the design of addressing and protection hardware, implementation languages, and operating system structure. His previous experience at Sperry Univac includes five years in their Defense Systems Division, where his assignments dealt with computer graphics, man-machine considerations, computer and data networks, general systems design, and where he was principal investigator for an R&D project on computer security. He has been an Adjunct Assistant Professor at the University of Minnesota, teaching courses in artificial intelligence, is a member of ACM, acts as a reviewer and referee, and has lectured at various conferences and workshops in both Europe and the U.S. He studied at Harvard University, receiving a B.A. summa cum laude in Physics and a Ph.D. in Applied Mathematics (Computer Science).

THE CHARGE TO THE GROUP

This session was to consider the vulnerabilities associated with the operation and maintenance of the central processor, operating system and hard-wire peripheral devices. Appropriate controls were to be considered from two different perspectives: the system design and acquisition phase and the ongoing system phase. [See Part I, Section 2 for the complete charge given to this group.]

The report that follows is the consensus view of this session.

Processors, Operating Systems and Nearby Peripherals

A Consensus Report

Theodore M.P. Lee (Chairperson)

Peter Neumann
Gerald J. Popek
Peter Tasker
Stephen T. Walker
Clark Weissman
James E. Rife (Recorder)

Note: This report was written by the chairperson but has been reviewed, revised and approved by all members. The views expressed represent the individual and collective opinions of the participants, and do not necessarily represent the views of their respective organizations, the NBS, the GAO, or of the sponsors of any work they have participated in.

OUTLINE

1. Purpose
2. Scope
3. Findings
 - 3.1 Lack of Problem Awareness
 - 3.2 Lack of Policy
 - 3.3 Lack of Technical Skills
 - 3.4 Inertia Problem
4. Recommendations
 - 4.1 Characterize the Problem
 - 4.1.1 State of Current Evaluations
 - 4.1.2 Vulnerabilities List
 - 4.1.3 Design Principles
 - 4.1.4 Technology Transfer
 - 4.2 Formulate Policy
 - 4.2.1 Aspects of DoD Policy
 - 4.2.2 Topics to be Covered
 - 4.3 Establish Evaluation/Accreditation Process
5. Evaluation/Accreditation Process
 - 5.1 Security Metric
 - 5.1.1 Overview
 - 5.1.2 Specific Features
 - 5.1.3 Architectural Features

- 5.2 Evaluation Matrix
- 5.3 Approved Products List
- 5.4 Administrative Aspects

6. Special Solutions

- 6.1 Periods Processing
- 6.2 Automated Periods Processing
- 6.3 Secure Distributed Processing
- 6.4 Secure Subsystems
- 6.5 Assurance of Special Solutions

7. References

1. PURPOSE

The task charged to our session was to list the vulnerabilities of the subject areas, and the counters to them, with some evaluation of costs. We decided fairly quickly that the general purposes of the workshop as a whole would be better served in a somewhat different way. We interpreted our charge as being better expressed as:

"What authoritative ways exist, or should exist, to decide whether a particular computer system is 'secure enough' for a particular intended environment of operation, and, if a given system is not 'secure enough' for an intended application, what measures could or should be taken to make it so?"

We were well aware that even beginning to discuss this question in a coherent way, much the less giving a complete and technically and administratively acceptable answer to it, is a formidable task. It was, however, the consensus that the state of the art is now such that the beginnings of an answer, in the form of both the technical steps to be taken and the administrative support for them, are close to reality. Accordingly, we formulated a number of steps, which are described here, and strongly recommend that they be considered by NBS and GAO as a program of action.

Although we appreciate the desire of NBS and GAO for a report that could form part of a Federal Standard, it is our conclusion that in the time available we could not directly prepare much that would be of use itself. Until the program recommended is substantially complete, the necessary information will not be available.

2. SCOPE

Our session was charged to deal with processors, operating systems, and nearby peripherals. Experience has shown hardware not to be a very important aspect of the problem. The significant computer security problem lies in any software that is supposed to fulfill a role in enforcing security. Accordingly, most of our discussion dealt with that problem. (The evaluation methodology to be discussed below does cover hardware, but that subject will not be discussed much.)

The major emphasis is on operating systems, but it must be recognized that other forms of software can and do play a critical role in security. Included in our scope therefore are also any special software subsystems, such as data management, transaction systems, or even micro-code, that are intended to perform security functions above and beyond those provided (or not provided) by the operating system.

3. FINDINGS

It was the consensus of our session that apart from certain elements of the Defense Department and the Intelligence Community, the federal government as a whole, not to mention private industry, is generally quite ignorant of most aspects of computer security. We find specific lacks and weaknesses in three areas, which are identified below, in sections 3.1 - 3.3.

We accept the definition of security as given by the tasking documents (availability, integrity, and confidentiality) but must further point out that security must be viewed as well from other perspectives. In particular, it is of fundamental importance that security involves a balanced attention to three subjects:

- o Policy -- the understanding and specification of what security rules, practices, doctrine, and administrative procedures are to be enforced for sensitive information, both in general for a given agency, department, or organization or class of information, and specifically for the role the computer is to play in enforcing them.
- o Mechanisms -- the choice, design, implementation, and use of suitable software (mostly) and hardware mechanisms to enforce the protection needed to support the desired policy.
- o Assurance -- the steps taken to convince whoever needs to be convinced that the relevant security mechanisms have been chosen, designed, implemented, and supported in such a manner that there is sufficient confidence that they do indeed enforce the chosen security policy in the face of the reasonable and credible threats they will be exposed to in a given operational environment.

Notice carefully that policy is not to be confused with mechanism [7], although the two are in practice closely related. A variety of sets of mechanisms are available that will support a given policy or set of policies [11]. The chosen mechanisms must be well enough matched to the policy that efficient, effective and natural support for it, including its administration, is feasible. The mechanisms must also fit cleanly into a rational software architecture so that the assurance tests can be met. On the other hand, the mechanisms must be viewed as implementation and protection tools to be designed according to the sound software and systems engineering principles of generality, comprehensibility, and robustness. For instance, it would be a poor choice of mechanism, although probably efficient and effective -- for a simple policy -- to tag every word of storage with an extra three bits that contained its security classification level, with special hardware checking the current security clearance of the processor (held in a special register) against the tag bits on every storage reference. (This example mechanism is a poor choice because it is too specific to a particular policy and does not really help assure the integrity of the software, tables, and high-level authorization mechanisms that would be used to set up the values in the tag bits in the first place. It only models the hierarchical aspect of a security policy and does not cover either the non-hierarchical aspects or the integrity aspects.)

Our specific findings identify the general state of affairs we see, either in the federal government or in the technical community, with respect to how well attention has been paid to these subjects and to the prospects for the future.

3.1 Lack of Problem Awareness

There continues to be a (to us) surprising lack of awareness that there is a "technical computer security" problem. It is a fact, demonstrable by any of several studies, that no existing commercially-produced computer system can be counted upon to protect any of its moderately knowledgeable users from having complete and undetectable access to any information in the system, no matter what kinds of so-called security features or mechanisms have been built into the system [1,2,3,8]. Despite this, government agencies, as well as

private industry, continue to issue purchase requests containing sections labelled "security requirements", which are mostly lists of features and mechanisms, in the apparent belief that they will obtain something useful.

It is true that mechanisms can be (and are) supplied that will prevent unauthorized people from using the computer, but once someone is allowed onto a system, the internal defenses can just not be trusted, no matter what their external appearances may be. Experience has shown that any existing system can be (and frequently has been) successfully "broken" with less than about three man-months of effort; in many cases, much less. This is even true for those systems to which serious "repair" efforts have been applied.

It must be pointed out that even the most recent releases of major vendors have been successfully penetrated. This includes those systems for which the vendors have conscientiously and competently attempted to improve the security, even as recently as in 1978 [23].

It should further be stressed that the skills needed to effect a successful penetration are not as arcane and scarce as some claim or wish to believe [8]. In fact, scenarios have been worked out whereby much of the labor to effect a penetration can be performed, under the direction of the true penetrator, by newly hired, newly trained people, such as recent computer science graduates, who would not need to be aware of the true purpose of what they are doing, including, for instance, the specific installation that is the target for penetration. Notice that although it might take some effort (estimated to be on the order of \$100-\$1,000 per flaw) to find a system flaw, once that flaw is discovered it can be repeatedly exploited on any similar system at little cost, and, that once a flaw is discovered, knowledge of it tends to be rapidly disseminated. (The added effort needed to develop the tools to exploit a flaw depends partly on the amount of information to be illegally gained or modified: obtaining a single password takes very little; printing several large files, including altering, avoiding, or removing traces of the clandestine activity, could take the full three man-months.) And, if the flaw is a manifestation of a fundamental design defect in the system, as many are, it may be prohibitively expensive to correct it.

Although these facts have been drawn to the attention of the government and the public on many occasions [25], there seems to continue to be a large body of people making uninformed decisions about security. The situation has not changed much since the earliest and most widely available reports on the subject, such as the Ware report of 1968 [20] and the Anderson report of 1972 [2]. Although these were written for DoD needs, they are generally applicable to any environment today.

In short, by any reasonable definition of "secure", no current operating system today can be considered "secure", nor are we aware of many under development that are likely to merit that adjective when they are done.

We hope the reader does not interpret this to mean that highly sensitive information cannot be dealt with securely in a computer, for of course that is done all the time. The point is that the internal control mechanisms of current operating systems have too low integrity for them to be able to effectively isolate a user on the system from data that is at a "higher" security level than he is trusted (or allowed) to deal with. For instance, in current DoD practice (somewhat simplified here) a given system at a given time is run at a single security level and external controls are used to ensure that only users cleared at least to that level are allowed on the system; at no time are multiple security levels of information handled on the system in such a way that the operating system has the responsibility of preventing a user from having access to information he is not entitled to. The system as a whole is run securely, and can be called "secure", but the question of whether the operating system itself is "secure" is simply avoided.

3.2 Lack of Policy

Except for the DoD and Intelligence Community no standard, well-thought-out, established policy about information security exists in the federal government. (And it must be admitted that the DoD and Intelligence Community policies are far from ideal, still evolving

with regard to computer security, and have as their biggest virtue at least the fact that they have withstood the tests of many years of experience with them.)

We observe that in most of the federal government it is not possible to talk about almost any matter having to do with information security in terms that would be fairly universally understood. There do not exist standard ways of categorizing and identifying sensitive information, nor do there generally exist rules and procedures for deciding who is allowed to have what kind of access to what kinds of sensitive information, nor are there rules for what practices they are supposed to follow in handling it.

This lack of a uniform, coherent policy about information security makes paying serious attention to the computer security problem difficult, because it is almost impossible to engage in a discourse on the subject using terms that have a good chance of being understood. For instance, as part of our deliberation, and contained in the recommendations below, we propose that an "approved products list" be established, whereby particular systems are identified as being approved for operation in particular environments, dealing with particular kinds of sensitive information. In the national security world, for example, we are able (in principle) to say that "Operating System 720, version 92.65, of vendor RUR (ADP Div.), running on hardware level 24, system 4545, is approved to handle mixed SECRET and TOP SECRET data provided that all users are cleared at least to SECRET and" and most involved will understand what all that means. In the rest of the federal government, there do not exist the vocabulary and practice to even begin to define the kind of sensitive information included in a given system, the nature of the environment it is to run in, the trustworthiness of the people, and the required physical, administrative, personnel, and communications security.

3.3 Lack of Technical Skills

It is our observation and finding that the technical skills needed to adequately analyze the security of a given system in a given environment, or to provide solutions to the problems that might be found in such an analysis, are not widespread. In particular, and perhaps this is more important, people in the federal government responsible for procuring computer systems generally do not possess the skills and experience to write the computer security portions of procurement specifications. As a consequence, vendors are either in the position of not taking government security needs seriously, or of responding to the letter of the specification with features and mechanisms that do not satisfy the true security needs of the procurement. The problem is compounded by a lack of government experience in analyzing the vendors' responses.

3.4 Inertia Problem

Although the beginnings of the solutions to the current technical security problems exist, there is an inherent inertia in the development and procurement cycles. The inertia has two components: a vicious circle of demand not well enough specified to promote progress; and the truly major, although we believe on the whole beneficial, changes to the software development process that would be required to have security that is much improved from what it is today. Part of this second component is the need for a transfer of technology between the research community and the vendors.

At present, customers generally have to accept what the vendor gives them. Conversely, there is little pressure from customers for security, mainly due to customer confusion as to what is available, what could be available, what is needed, and what it wants. We observe that, generally speaking, users of computers do not in practice care very much about security, although this should change as more highly integrated networks and data bases come into widespread use. Customers tend to wait for the product while vendors are waiting for an indication of demand. This passive attitude on both sides tends to mask the general nature of the security problem because the more knowledgeable security users demand solutions for their unique problems, solutions that might not be of general utility and hence do not become standard parts of a product line.

Lack of attention to the computer security problem is also caused in part by a widespread belief that changing technology will make the problem go away. It is claimed (or hoped)

by many that as hardware costs decrease, each user or small group of users will have its own computer, and hence there will be no computer security problem; in short, the magic phrase "distributed processing" is often viewed as a solution to the computer security problem. Although there are cases where this is true (e.g. replacing a conventional time-shared service bureau with a lot of mini-computers), the major purpose of distributed systems -- the ability to widely and selectively share information -- requires that a distributed system at a high level of abstraction appear to be a single integrated system. And securely controlling access to and within this single system, whether distributed or not, comes up against the general computer security problem we have been discussing. Distributing processing hardware may remove the problems caused by a desire to share hardware resources, which is what has given rise to current perceptions of the computer security problem, but it does not remove the problem caused by a need to share information and the logical means for accessing it. See section 6.3 for a further discussion.

The technology transfer problem can be seen in the fact that even if government procurement specifications were tightened to ask for the kind of security we believe possible with the current state of the art, fewer than fifty people in the country would understand the true implications of what is being asked for, and those fifty are concentrated in less than a half-dozen organizations, none of them in the main-stream development organizations of the major mainframe vendors. This is partly because at the moment most efforts of vendors relating to security are concentrating on the "mechanisms" part of the security problem, with very little attention to the "assurance" part. The difficulty of technology transfer is compounded by the fact that the development of a new operating system, which is in effect what improved security is going to require, is a five to ten-year process, and if new software development and management tools need to be integrated into that process, the chances of incorporating them at the proper part of the cycle are not very good.

4. RECOMMENDATIONS

In our findings we have identified a number of problems and deficiencies in the current posture of computer security as it affects the federal government as a whole. It is our general conclusion that it would not be very meaningful to write a comprehensive set of guidelines until progress had been made in all these areas. Accordingly, we strongly recommend that the National Bureau of Standards, possibly in concert with other agencies such as the General Accounting Office, charter a group or groups of technical and policy people to specifically remedy the situation by performing the tasks recommended here. We acknowledge that this sounds like a perfect example of a committee's first job being to form a new committee, but feel it justified. We further note that the composition of these groups is important and that their members will have to be drawn from several sources, appropriately funded, and given a formal charter and direction. Expertise of those outside NBS is required.

The specific tasks that we recommend be performed are:

- o From available literature and people's experience, prepare a series of reports that characterize the current state of the art, including both the state of the technology and the state of current systems.
- o Formulate a detailed security policy, including especially nomenclature and marking schemes, for any and all sensitive information not covered by the relevant national security policies and guidelines.
- o Establish a formal security evaluation and accreditation process, including the publishing of an "approved products list", to guide specification and procurement of systems intended to handle sensitive information.

4.1 Characterize the Problem

Our first recommended task, which has four subtasks, is that a report or set of reports be prepared that make the state-of-the-art be available in a more accessible and collected form than currently exists. It is our observation that the computer security problem has been sufficiently, adequately, and voluminously documented already (see Carlstadt [5]), but that the available material is scattered in a variety of places, is varied in quality, is generally unevaluated (so that the inexperienced have a hard time deciding what is meritorious), and is frequently not very accessible.

Some relevant material, although we believe not very much, is either classified, proprietary, or for official use only. We recommend that every step be taken to sanitize, declassify, or otherwise make accessible as much of this material as possible. Some material, particularly that involved with penetration exercises, has never been written down and exists more in the form of the folklore of computer security. When appropriate, efforts should be taken through personal interviews or correspondence to extract this material.

The specific subjects to be covered in characterizing the problem are outlined below.

4.1.1 State of Current Evaluations

To deal with the lack of awareness of the nature of the computer security problem, and its reality, discussed in section 3.1 above, we recommend that the results of all past efforts to penetrate and repair operating systems be assimilated into a single report. The purpose here is only to broaden awareness of the problem, not to measure one system against another, nor to attempt to solve the problem. For this effort to serve its purpose it must, however, employ great candor and identify specific techniques used to break specific systems. Without this, the report will not be sufficiently credible to perform the necessary consciousness-raising function.

We acknowledge that undertaking this task will take great courage. Creative discretion will have to be employed so as not to reveal too many weaknesses, although we counsel disclosure rather than protection. A conscious effort should be made to qualify the seriousness of the known (or supposed) weaknesses of any given system or class of systems, but problems should not be minimized.

In addition to the documentation of the particular stratagems (preferably with actual code sequences) used in particular cases, it is perhaps even more important to document the kind of effort, level of knowledge, resources used, and history of each particular penetration exercise. (The MULTICS analysis [8] is a good example.) Also relevant are any cases where attempts have been made to repair a system, along with the outcome.

4.1.2 Vulnerabilities List

The experience gained from the past penetration exercises has generated a body of knowledge about the general kinds of vulnerabilities found in current operating systems. Example lists of such vulnerabilities are found in Bisbey [4], Linde [10], and Neumann [14]. Those lists and other similar ones from the literature or from personal experience should be pulled together to form a general characterization of the problem. Again, the purpose here is not to fix any given system, nor to help someone attempting to improve his security, but merely to document the state of affairs.

4.1.3 Design Principles

Over the years a number of design principles for security in operating systems have been proposed. Examples are found in Saltzer [17] and Neumann [14]. To guide future designers, and to serve as a checklist for those examining current systems, these lists should be collected together and merged. It should be noted that although following these lists will eliminate (or would have eliminated) many of the vulnerabilities to be identified under 4.1.2, above, not all will have been taken care of.

4.1.4 Technology Transfer

As discussed above in 3.3 and 3.4, the knowledge of what it takes to truly improve the security of a system is not widespread. Current research efforts, notably the KVM/370 (Kernelized VM/370) [6], KSOS (Kernelized Secure Operating System) [21], UCLA Secure Unix [24], and PSOS (Provably Secure Operating System) [15] projects -- as well as the MULTICS GUARDIAN effort [22] -- have developed some experience in using the kinds of software tools and management approaches that seem necessary. To aid the transfer of this technology, a report should be written that carefully documents the experiences of each of these efforts. The report should cover what the goals of each project were (or are), what kinds of tools were used, the experience -- especially learning curve -- in using them, the costs involved, and the results. Costs should include both the expense of the effort and the effect on the product -- performance and compatibility. The point here is not so much the detailed technology involved -- as that has to be covered by other means -- but a feel for "what it really takes to make a system secure."

4.2 Formulate Policy

Our second recommendation is that a group should be chartered to formulate security policy, practices, and doctrine for those parts of the federal government that do not already have them. We strongly recommend that this group examine the current practices within the national security arena (DoD and Intelligence Community) as a model. Under the provisions of OMB Circular A-71 it would appear that either OMB or GSA is the responsible authority for such an activity, but the technical recommendations and guidance of NBS would be necessary ingredients to the formulation of a policy. The other reports from this workshop may well have other suggestions for the administrative seat for such an activity.

In recommending that the DoD policy be considered as a model, it is our consensus that protection mechanisms suitable for supporting that policy are general enough to support any reasonable policy defined for agencies outside the DoD. Furthermore, it is also our consensus that mechanisms inappropriate for the DoD policy will also be found inappropriate for most other generally-applicable policies.

We recognize that in many areas the DoD policy is more constraining and inflexible than would be desirable or feasible for other arenas, even within the federal government. For instance, it will probably be found necessary, especially in the more commercial uses, to distribute the right to selectively downgrade information much more widely than is allowed in DoD practice. How such a right can be adequately controlled and audited are administrative and technical issues for which current DoD practice does not establish a good precedent. Some effort needs to be spent on formulating policy in these areas, and the evaluation criteria proposed below may need expansion to cover them. We caution that the formal work on DoD security policy (as found in some of the references mentioned in section 4.1.4, above) has concentrated mostly on protection and access control issues and has not covered the broader issues of security administration very well.

4.2.1 Aspects of DoD Policy

There are two aspects of the DoD computer security policy, as it has evolved recently, that are not widely understood but which we strongly recommend be given consideration in the formulation of non-DoD policy. The first is the distinction between discretionary and mandatory access control and the second is the distinction between a hierarchical (security levels) and a non-hierarchical (lattice-structured security compartments or categories) expression of mandatory access rules.

When information is given a formal security classification, it is forbidden without explicit administrative declassification or downgrading to allow someone to have access to information of higher classification than he is cleared for, i.e., the holder of classified information has no discretionary authority in this respect concerning who he can share it with. This rule is an example of a mandatory (also called non-discretionary) access control policy.

Enforcing this rule in a computer is in a way more of a concern than in the manual world of printed information. In the manual world, the holder of classified information always has direct physical control over the information (except when it is stored somewhere or in transmission, which is where physical and communications security come into play.) The security of the information thus depends almost solely on the integrity of the person holding it, which is where personnel security (e.g., clearance and indoctrination procedures) comes into play. In the computer, there are always one or more pieces of software (and hardware) interposed between the person and the information. It is infeasible to arrange that all of this software is completely trusted to carry out the person's wishes. Unless the security policy is enforced by the core of the operating system over the rest of the system (e.g., word processing software, scientific routines, compilers, control language interpreters, communication handlers, game playing programs, and data management systems) there is always the real chance that an untrusted piece of software will violate security, either through error or through (undetected) malicious clandestine intent.

The possible violations could be a simple copying of information into a place accessible to someone not authorized to receive it directly, changing its classification, or covert signalling of it through a variety of technically sophisticated but very real mechanisms [9]. In particular, an untrusted (i.e., unaccredited) piece of software, even though written by a trusted person and acting under his control, must not be allowed to read files containing information at a higher security level than any files it is allowed to write into, lest it accidentally or maliciously downgrade the information by copying it into the lower-classification files (which could have been set up exactly for that purpose by a would-be-information-thief.)

In the real world sensitive information is not neatly organized into well-ordered sets of increasing sensitivity, and the security policy enforceable by a computer should accommodate this fact.

For example, a person's medical file and his financial file in an organization both contain sensitive information, but neither can be said a priori to be more or less sensitive than the other, nor does being allowed access to one imply anything about being allowed (or not allowed) access to the other. To handle analogous situations the national security community uses a mandatory security policy that involves both a notion of level of sen-

sitivity and a notion of arbitrary grouping of sensitive information (loosely referred to as compartments or categories.)

Thus one could say the medical file belongs to (is labelled with) the category MEDICAL, and the financial one, FINANCIAL. An organization's physicians and nurses would be cleared for (labelled with) the MEDICAL category, but not the FINANCIAL one, and the payroll department and certain management personnel would be cleared for (labelled with) the FINANCIAL category. A few highly-placed management people might also be cleared for (labelled with) both categories.

To enforce a mandatory policy on such information, if a person -- perhaps the organization's psychiatrist -- wishes to read both a MEDICAL and a FINANCIAL file at the same time (perhaps to merge extracts from them into a common report) he would not be allowed (at that time) to write into a file of either category (lest untrustworthy software "accidentally" change the classification of, say, MEDICAL information to FINANCIAL, by copying it into a file labelled FINANCIAL, thereby allowing the payroll department access to medical information.) He could, however, create a special file labelled with both FINANCIAL and MEDICAL and freely copy from either category of information into it (but not the other way around.)

In addition to a mandatory security policy, systems must also support a discretionary policy wherein each creator of information can say who is allowed to use it, and in what way (e.g., for reading, writing, appending, or executing as a program), within the constraints of the mandatory policy. This discretionary mechanism is what most systems have today and it will continue to be an important way for people to go about their business. In the DoD it is the way of expressing "Need-to-Know" -- to have access to classified information a person must not only have the proper administrative clearanc(es), but also the owner, possessor, custodian, or administrator of the information must determine that the requestor really needs the information for the purpose at hand. Notice that most of the implied security rules of the various privacy acts, regulations, and practices seem to be of this latter discretionary variety. However, the rules of the IRS, for instance, and the rules governing inter-agency exchange of information under the Privacy Act seem to be more of the mandatory kind. Imposed on top of the mandatory categorization scheme of our example then there would also be a discretionary scheme that, for instance, did not allow all management people access to all financial information, but perhaps only to that of their immediate subordinates.

The MEDICAL/FINANCIAL example we have used here is admittedly a bit forced (since it is unlikely that most people would consider the possible threats to their privacy severe enough to require the rigid kind of policy suggested), and over-simplified (there should at least be a difference between what the nurse and the physician can do), but we hope it illustrates the kind of situations that do occur. To make the example more realistic, the reader could imagine that the system containing the medical and financial information also contains information that is the property of two activities that must be kept separate from each other, although a few people are allowed access to information about both activities, and some information pertains to both activities. (In a commercial environment the two activities might be internally competitive efforts; in other environments, such as government, they might be negotiations with opposing sides on some highly controversial matter.) A further complication to the example would arise when implementing the need for the physician and employee to both have access to the employee's file, but in different ways: the physician could alter it, but the employee could at most read it and add comments to it without changing existing information in it. Current privacy legislation requires exactly this kind of facility. The major point of our example is that a useful, commercially viable security policy must be quite general (just security levels is insufficient) and it must be easily customized to a particular environment -- when our example said the files had security labels "FINANCIAL", "MEDICAL", or both, it meant what it said.

4.2.2 Topics to be Covered

Included in the policy to be formulated should be attention to the following matters:

- o What kinds of sensitive information are to be covered by the policy.
- o Who has responsibility and authority over each kind of sensitive information.
- o What kind of access control and record-keeping are to be applied to each kind of sensitive information.
- o What are the rules for both discretionary and mandatory access control.
- o What are the rules for exchanging information between different responsible agencies (the privacy act already covers some aspects of this.)
- o A federal-government wide standard for marking and identifying information that is sensitive for other than national security reasons.

As we found above in section 3.2, attention to these matters is required before it becomes possible to talk meaningfully about what kind of security policy is to be enforced in a computer system, "how good" the security should be for a given application, and whether a particular system is "good enough" for a particular application.

We cannot stress too much the requirement for a standardized marking scheme, as burdensome and fraught with bureaucratic tangles as it may seem. Such a standard is required simply as a matter of good security practice -- it is important that each person dealing with sensitive information be put on notice that he is dealing with a particular kind of sensitive information, and this is best done by establishing a standard set of markings, each marking conveying to all who see it a sense of the rules associated with handling that information. Furthermore, once such a set of marking and identification rules is established, it becomes meaningful to talk about what role the computer is to play when dealing with each kind of sensitive information, and for the system to automatically take care of some of the administrative burden.

4.3 Establish Evaluation/Accreditation Process

Our third and final recommendation is that steps be taken to formulate and institutionalize a process for evaluating the security of computer systems, and for accrediting particular systems for particular applications. We recognize the administrative and political problems inherent in such an endeavor, not to mention the technical difficulties to be encountered. We do however believe that major strides can be made. Since this is our major recommendation and since it would be a major effort, section 5 below is devoted to a detailed description of what we believe such a process could (and should) look like.

In addition to the general development of an evaluation/accreditation process described below, we recommend that two preliminary steps be taken:

- o That a standard set of procurement specifications for computer security be developed. The form of the standard might be as a collection of paragraphs that could be selected from and tailored for each individual procurement. A start for such a standard can be found, for instance, in the SATIN IV (SACDIN) [12] and KSOS requests-for-proposals. A main purpose in having a procurement standard would be to focus design and evaluation efforts. The standard should encourage early evaluation of design prior to implementation, e.g., through proofs that specifications satisfy formal requirements. And, it should encourage the preparation and retention of suitable documentary evidence throughout the design and development process.
- o That a preliminary evaluation of the more popular or security-critical current systems be performed and possibly an "approved products list" be published. The basis for the evaluation would be the security metric discussed below in 5.1, except that the evaluation would not be as formalized (mostly narrative description) as possible later.

5. EVALUATION/ACCREDITATION PROCESS

In addressing the issue of "how good is the security of my system", words like "trustworthiness", "certification", and "accreditation" must be given substance. In DoD applications, certification is a technical process that examines risks, expected losses that might obtain from a given set of threats, and the effectiveness of the enforcement mechanism to counter those threats. The technical assessments generate a set of "evidence" documents which may include risk assessment tradeoffs, security architectures, program development and testing, and formal proofs of correctness.

Accreditation is a management judgment that the evidence is credible, and sufficient to support the contention that the enforcement mechanism counters the known threats. The mechanism is then deemed trustworthy and approved to operate in the specified manner. Accreditation involves technology for generating credible evidence and for assessing the adequacy of the technical measures employed. It also involves establishing policy regarding acceptable levels of risk. In non-DoD government activities, a most serious weakness, as we have observed, exists in the absence of a sound, technical security policy. In this section we recommend a series of measures to remedy these problems.

Vendors have observed, with good reason, that there exists a great amount of confusion, inconsistency, and technical immaturity in the way the government (and others, as a matter of fact) includes computer security in its procurements. Part of the reason for this, as we observed above, is the shallowness of the technical bases to draw upon. Accordingly, we recommend that a standard, formalized, institutionalized process for dealing with the data security parts of procurements be established along the lines to be discussed here. We note that the proposed process is also being considered by the DoD (under whose auspices some of the concepts originated) and hence there may be an opportunity to combine resources and experience.

The suggested process has both a technical aspect and an administrative aspect; we will not dwell much on the administrative aspect, although we recognize it to be of great importance to ensure that the process is accepted, is fair, and works well.

The technical process has three parts, which are presented in further detail below in sections 5.1 - 5.3:

- o A technical means of "measuring" or evaluating the over-all security of a system.
- o An evaluation matrix that identifies how "good" a system should be to be approved for operation in a particular kind of application in a particular kind of environment. The key idea here, which requires the formalization of the non-DoD security policy outlined in 4.2 above, is being able to identify what "level" of sensitive information a system will deal with, what kind of freedom its users will have, and what threats to security must be countered by the system because they are not dealt with by physical, administrative, personnel, or communications security.
- o An "approved products list" that identifies which particular systems are approved, under the security evaluation matrix, for operation in each of several major categories of operational environment.

We are fairly confident that at least an identification of the elements of the security evaluation can be evolved in a technically sound and unambiguous way; the other steps are much more dependent on the as-yet-unavailable understanding and formalization of the kinds of sensitive information to be encountered across the federal government.

5.1 Security Metric

A metric is a repeatable, unambiguous measure of some attribute of an entity such that the measures for two different entities can be compared in a meaningful way. At present the security "attributes" of two different computer systems can generally only be compared intuitively. We believe it possible to capture the underlying principles used in such an intuitive measure into a formalizable set of values for a "security metric", along the lines discussed here. We admit that the development of this metric is far from complete and will be subject to some amount of disagreement, but on the whole even the first cut described here will be generally acceptable, capable of sufficiently rigorous and unambiguous application, and certainly a good step towards rationalizing the process of security evaluation.

Since few of the relevant security attributes of a system have numerical measures (one cannot meaningfully ask "what is the amount of effort needed to break into system XYZ"), we propose instead a hierarchical list of features to be looked for. The list of features has a major division into those representative of the technical mechanisms for enforcing security in a system and those representative of the assurance one has about the efficacy of those mechanisms. The structure of features to be examined can be portrayed as an "inverted bull's-eye", as shown in figure 1, where the farther a feature is from the center, the better the security is (in so far as one aspect of the system is concerned.) Generally speaking, an outer (or "better") feature of a given attribute cannot logically be present in a system without the inner (or "poorer") ones also being there. When this is not the case, no credit can usually be given for implementing a "better" feature without also implementing the ones inferior to it.

Rather than measuring a single "attribute" of security we are in fact proposing that several different attributes, each of which has a bearing on security, be measured. (Intuitively, all the features measuring mechanism could be "added" together to form one measure, those for assurance another, and then the two measures "multiplied" to form an over-all measure. But this is a simplistic generalization, since no amount of some measure may be able to compensate for an insufficient amount of another measure. In particular, practically none of the prevention or detection features are worth anything unless at least the first non-null steps in most of the assurance attributes are present. For instance, if the prevention mechanisms are ineffective, the detection and authorization mechanisms, no matter how good they seem, can be bypassed.) As a start, to specify the required security for a given application, or to characterize a given system, it will suffice to list the relevant features.

5.1.1 Overview

The evaluation of a system would be based on one extrinsic attribute (policy) and on two sets of intrinsic attributes (mechanism and assurance.) Although the metric outlined here only measures the intrinsic attributes, that measurement must be done in the context of the external requirements upon the system. The primary categories of the evaluation process then are:

- o Policy -- the explicit security policy the system must enforce. Without its being stated and clearly understood there is no way of evaluating whether the mechanism implements its specifications, or that the specifications satisfy the security requirements.
- o Mechanism -- the specific features and mechanisms intended to establish a high-integrity protection environment to support the stated security policy, either directly by controlling access or indirectly through various administrative tools.

* Prevention -- the mechanisms, mostly software, that are intended to prevent breaches of security; also called enforcement mechanisms. Should be sufficiently powerful, general, and efficient to counter all relevant threats.

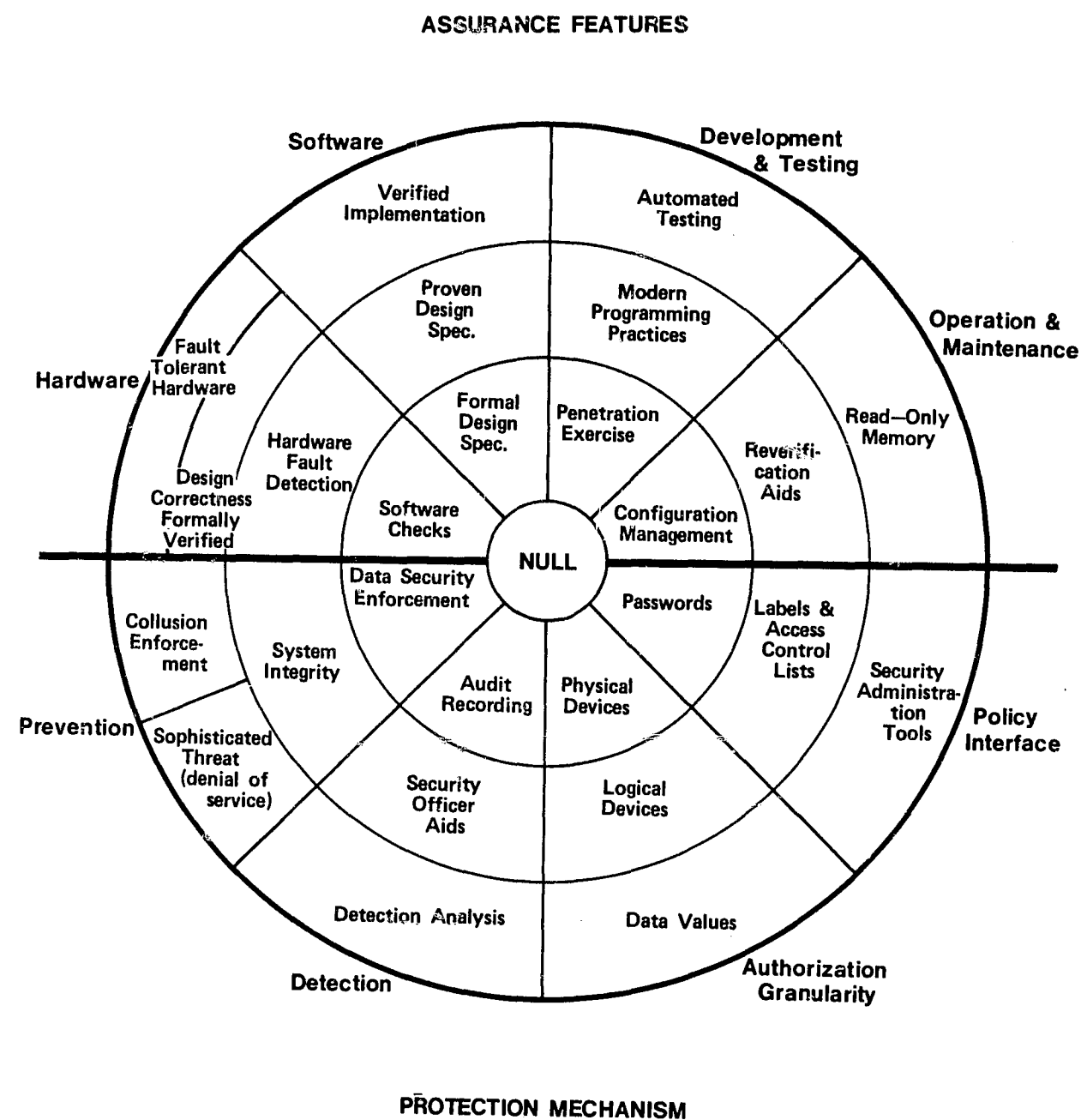


Figure 1 - Security Metric

* Detection -- the mechanisms to detect suspicious events, or to provide a security officer with on-line and off-line security assessment tools. Should permit rapid detection and confinement of error propagation.

* Authorization -- the mechanisms to authorize accesses and manipulations of a system object to a user (or program), including administrative support for them.

- Authorization Granularity -- to what level of detail accesses can be controlled.

- Policy Interface with Authorization Mechanisms -- the manner in which authorizations are expressed and decided vis-a-vis the stated policy.

o Assurance -- features that measure the confidence one has in the security mechanisms. Covers both the proper design of the mechanisms and the assurance that they operate correctly and reliably in the face of both malicious intent and accident, including failures.

* Hardware -- assurance that the security related hardware operates correctly.

* Software -- assurance that a formally stated security policy is supported by the software system design and implementation (software includes relevant firmware; a more detailed metric would have separate measures for firmware).

* Development and Testing -- assurance that the system is developed using formal and rigorous controls and standards.

* Operation and Maintenance -- assurance that the system is operated and maintained in such a manner as to warrant continued confidence in it, even in the face of vendor or user enhancements, modifications, or repairs.

5.1.2 Specific Features

The extent to which a given system provides security then is measured by looking at what features it possesses for each of eight attributes, four for assurance, four for mechanism. The possible features for each attribute generally form an ordered list, where improved security is to be had by including successively "better" measures. Each feature represents a small range of security improvement, where in order to be more precise it is necessary to say to what extent or how well that particular feature is represented. After each feature is given a nominal figure to represent the contribution to security (in the given attribute dimension) that presence of the feature represents. The numbers picked are somewhat arbitrary, except that increasing numbers mean better security and features with considerable range of implementation are given a range of numbers. There is no intention that the numbers in one dimension be comparable to those in another, and it is not meaningful to say that a figure of "8", say, is twice as "good" as a figure of "4".

For the eight attributes then, we have the following possible features:

1. Prevention

a. Null -- system incapable of enforcing security and integrity principles, either through lack in hardware mechanisms or through lack of software features. (0)

b. Data Security Enforcement -- system intended to be capable of enforcing basic security principles (unauthorized direct access and manipulation)

upon individual users attempting (accidentally or maliciously) to directly violate the security policy; system does not seriously intend to defend against the more advanced forms of indirect programmed attacks. (1-3)

- c. System Integrity -- system intended to be capable of protecting its own integrity against malicious or accidental acts of its users, including surreptitious programmed attempts to bypass or fool the protection mechanisms. (4-5)
- d. Collusion Enforcement -- system intended to be capable of enforcing data security with two or more users colluding (e.g. using covert channels) including the threat of information leakage via implanted software (unintentional downgrading). (6-7)
- e. Sophisticated Threat -- system intended to be capable of enforcing security principles involving sophisticated threats including the more complex denial of service threats. (8-9)

Note: these last two features are not strictly ordered and are representative of the most sophisticated kinds of mechanisms that can be implemented and the threats they defend against.

The evaluation of the prevention attributes is based solely on the intent of the mechanisms furnished, possibly on their efficiency, but not on how effective they are. That measure is taken under the assurance attributes.

2. Detection

- a. Null -- system incapable of detecting misuse (to any great extent) in real time and no facility for detecting or assessing damage after an unauthorized penetration. (0)
- b. Audit Recording -- system capable of logging critical security actions in sufficient detail to provide an adequately complete trace of user and system activities. (Measure of compliance dependent on the extent of the records and whether they are easily directed to a cognizant person for analysis; for instance, if the threat is an authorized user turning bad and misusing his legitimate authority, only the owner of a data base can help by ascertaining whether accesses to it by each person are really necessary and proper.) More effective if audit is on non-modifiable medium (to prevent after-the-fact tampering.) (1-4)
- c. Security Officer Aids -- measures to help security officer monitor in real-time system activities, even to the extent of transparently watching users activities. More credit given if audit trail is accessible (read-only) to security officer via a modern data base management system with good on-line query capability. (5-7)
- d. Detection Analysis -- enhanced on-line security facilities and sophisticated post analyses to perform history/trend and pattern recognition analysis. (8-9)

Note: the rating of the detection features of a given system would be affected by the prevention rating, since an audit record that is not data secure might be worthless or even misleading.

3. Authorization Granularity

- a. Null -- system incapable of enforcing access controls. (0)

- b. Physical Devices -- system can enforce control over access to physical devices. (1-3)

- c. Logical Devices -- system can enforce control over access to logical data structures. Measure of compliance increases with finer granularity, ranging from Files, to Records, to Fields within records. (4-6)

- d. Data Values -- system can authorize based on the value of the data element being accessed, or on the value of some other data element. More credit if system can also perform legality checks (as implied by the security policy), threshold checks, and reasonableness (e.g., type or range) checks. (7-9)

We recognize that this analysis procedure would need more refinement to cover subsystems with the more elaborate protection structures, such as data management systems, but the general principles would apply. (See [16] and [19].)

4. Policy Interface

- a. Null -- system has no ability to selectively enforce accesses in accord with stated policy. (0)
- b. Passwords -- system provides only for passwords as a means of controlling access, either to the system as a whole or to data within it. (We assume that generally speaking passwords are always used as the means of authenticating a user on entrance to a system, recognizing that other schemes are possible and at times permissible or preferable.) (1-2)
- c. Labels and Access Control Lists -- system provides for selective control of access to data based on the specified security properties of data. Whether distinction between discretionary controls and mandatory controls is accommodated, and how mandatory controls are expressed would be taken into account in a particular evaluation. Best score obtained when at least the full lattice structure (levels plus categories/compartments) of the DoD policy is or can be effectively implemented. (3-6)
- d. Security Administration Tools -- appropriate tools are provided to make administration of access control more convenient and subject to audit and automation; also included in the measure here is the appropriateness of the division of labor between the user, owner (or custodian) of data, and security officer/administrator. (7-9)

5. Hardware Assurance

- a. Null -- system has no facilities for verifying that the hardware is working correctly. (0)
- b. Software Checks -- system has software that checks the integrity of the security related hardware periodically. (1)
- c. Hardware Fault Detection -- system has hardware that detects an incorrect operation of the security-related hardware. More assurance achieved if hardware verified to fail safe than if hardware can fail unpredictably. (2-3)
- d. Hardware Design Correctness Formally Verified -- security related hardware proven and verified to be correct in operational and degraded environments. (4-6)
- e. Fault Tolerant Hardware -- hardware designed in accord with high reliability standards akin to those used in deep space probe missions or nuc-

lear reactor control; important if system availability is a major goal for a given system. (7-8)

6. Software Assurance

- a. Null -- system whose design specification is not formally and rigorously stated and the resulting implementation is developed and tested using traditional testing techniques. (0)
- b. Formal Design Specifications -- system specified via a rigorous specification language. The implementation may also be implemented via verifiable procedures, but the design has not been formally proven to be a correct and complete representation of the specification and the code has not been verified. (1-3)
- c. Proven Design Specification and Verifiable Implementation -- system developed using provable techniques (assisted by automation) for ensuring the design accurately reflects the design specification. The implementation is in principle verifiable, i.e., the proper rigor has been taken and proper language used, but it has not been formally verified. (4-6)
- d. Proven Design and Verified Implementation -- system developed using provable techniques (assisted by automation) for ensuring the code accurately reflects the design and specification. (7-9)

7. Development and Testing Assurance

- a. Null -- system developed without complying with formal and rigorous controls and standards, and no special attention has been paid to security aspects of the implementation. (0)
- b. Penetration Exercise -- system has been subjected to a thorough attempt to penetrate its defenses, any discovered design or implementation weaknesses or flaws have been corrected, and the process has been repeated until no further flaws are discoverable. (1-2)
- c. Modern Programming Practices -- system developed via a carefully controlled and managed implementation process to include software management, structured walk-throughs, top-down programming, structured programming and testing, etc. Implemented with a modern high-level language (structured assembly language, FORTRAN, COBOL, etc. are insufficient.) (3-5)

Note: this feature is sufficiently rich and important as to probably merit its own sub-evaluation criteria.

- d. Automated Testing -- in addition to the above, the system has been tested in a semi-automated way against a test specification proven to be an accurate representation of the design specification; test must demonstrate complete implementation of the specification and the validity of all assumptions, not merely that the external interfaces work as intended when used as intended. (6-7)

8. Operational and Maintenance Assurance

- a. Null -- no particular attention has been paid to the state of the security aspects of the software after delivery. (0)
- b. Configuration Management -- rigorous controls over the software and hardware configuration are employed after system is operational, including careful bookkeeping and authorization of changes, with at least a comparison of proposed changes to the security specifications. (1-2)

- c. Reverification Aids -- any changes are re-verified with the tools and to the level of formalism used in the design and development process; better if automated tools help keep track of what parts of the system need to be re-verified for any given design or implementation change. Unannounced, aperiodic "Inspector General" visits used to check the system, its controls, and operations. (3-4)

- d. Read Only Memory -- security relevant trusted software (including all of the "security kernel") is run from read-only memory, distributed and controlled by a central authority for a given system. (5)

Note: All security-relevant copies of code and design specifications that will affect what software is run on the machine must be protected with sufficient care to assure their integrity to a level commensurate with the level of sensitivity of information to be processed, if unauthorized changes in them can (or could) be used as vehicles to defeat the security enforcement. This fact should be considered in evaluating both 3 -- Development and Testing Assurance -- and 4 -- Operational and Maintenance Assurance. A software configuration integrity policy will have to be developed as part of the security policy formulation recommended in section 4.2. One current DoD practice is to subsume security-critical software under classification practices, which is not strictly correct since usually reading such software is not prohibited, only modifying it.

5.1.3 Architectural Features

It is not, and should not be, the role of the proposed evaluation/accreditation process to specify or recommend specific sets of mechanisms (hardware or software). To do so would at the least be an improper and probably illegal role for the federal government (telling vendors how to build their products), would tend to stifle innovative competition, and would certainly be difficult to place on a firm technical base. For instance, in the DoD context it would be legally improper and technically inadvisable for a federal standard to say that a computer must possess a tagged, capability-based architecture to be accredited for DoD multi-level secure operation at level 6 [see section 5.3], even if possessing such an architecture might be viewed by some as highly desirable.

On the other hand, the specific mechanisms present in a given system would certainly be taken as pieces of evidence in the over-all evaluation. Present experience, limited though it may be, has already identified a number of clearly desirable mechanisms and approaches. This section summarizes these and indicates where in the evaluation process they are relevant; that a given system did not use one or more of these mechanisms would not necessarily disqualify it for operation in a given severe environment, nor would their use automatically qualify it, but their presence would at least indicate that the system's designers are familiar with the state of the art.

- o Software -- The over-riding concern in software is whether it measures well along the assurance dimensions. The more severe the assurance tests, the less freedom there appears to be in the choice of a viable software architecture. At the least, in order to move out of the "Null" region there must be the recognition of a "high integrity" critical portion of the operating system software that establishes and maintains a general protection environment. There should be a carefully designed part of the operating system that is solely responsible for establishing and maintaining a set of protected environments for use by all of the other software -- OS and user -- and, furthermore, it is responsible for nothing else. This part of a system has come to be known as the security kernel. It provides general protection for a general-purpose processing environment; different protection policies (in support of different security policies) can be moderately easily specified and imposed once this basic mechanism exists. Notions of hierarchical decomposition, levels of abstraction, per-process virtual environments, software capabilities, and the principle of least privilege are all valuable. To be avoided are ad hoc mechanisms that intuitively seem to improve security but which in fact are not clearly

motivated by nor derived from an explicit statement of a security policy. Most important is that the chosen architecture can be and has been formally specified with rigor.

o Hardware -- Once features are available to provide a basic isolation between users and between user and system, the primary motivation for hardware features in support of security is the ability to efficiently implement the chosen software architecture and features. Any hardware holding security-critical information (such as current privilege mode) must be sufficiently reliable that single hardware failures will not disable protection. If any of the formalization steps in the assurance dimensions are taken, it will be found important that the hardware architecture be clean enough to be easily included in the formal specifications (and any subsequent correctness proofs.) At the least, hardware should support multiple execution contexts (protection states) with fast switching between contexts, e.g., process switch, domain crossing, procedure call. Virtual memory, tagged memory, capability registers, and descriptors are all valuable tools. And, of course, the instruction set must be partitioned into privileged and non-privileged subsets, preferably with several -- not necessarily hierarchically related -- privileged subsets.

5.2 Evaluation Matrix

Once the security metric has been worked out, it is possible to begin to speak meaningfully about how "good" a system needs to be for a particular environment. We propose and recommend that a "security standard" in the form of a three-dimensional matrix be prepared that identifies how "good" a system must be to operate in the possible applications and environments of interest. The dimensions of the matrix would be as follows:

1. Threat Environment -- the nature of threats to security the system is exposed to by virtue of the physical, administrative, personnel and communications security measures to be applied to it.
 - a. Accident/Browsing -- all users are supposed to be trustworthy, or are sufficiently constrained, that the only threats are accidental disclosures of information or discovering of unauthorized information through browsing.
 - b. System Exploitation -- users' trustworthiness is unknown or variable, and users with sufficient knowledge and skills can have a sufficiently un-constrained access that there is a significant risk that they may attempt to exploit system weaknesses.
 - c. System Modification -- either through system exploitation or through subversion of the development process, there is a significant risk that the system itself will be modified (so as to make unauthorized accesses possible.)
2. Application Environment -- the nature of the applications to be run on the system; if different users will be performing different applications, several combinations of threat environment and application may be involved.
 - a. Simple queries -- user can only ask to receive small amounts of data, through a simple query language. Cannot modify (much) information.
 - b. Simple Transactions -- user is still constrained to operating through a simple, formalized interface, but can generate transactions that modify data. (A bank teller or reservations system is a good example.)

- c. Application Subsystems -- user is constrained to operate within a

special-purpose application, perhaps one including a simple programming language. (But must beware that the application is simple enough, and controls strong enough, that he cannot break out of it.)

- d. User programs -- users can write and run arbitrary programs in assembly language; constraining them to a less powerful language may help, but should in general not be counted upon to lessen the problem.

3. Level of Data Sensitivity -- the most sensitive information to be handled by the system (at the time the least trustworthy user running in the least constrained application is allowed on). The entries in this dimension cannot be filled in until the policy formulation of section 4.2 above is carried out.

Once the dimensions of the matrix have been determined, a "panel of experts" would fill in for each meaningful combination what set of values of the security metric is required to permit operation of a system in that environment. Notice that a given system may occupy several of the intersections of the matrix since users of different degrees of trustworthiness may be using applications under different constraints. The most severe set of values of the metric would then apply.

We admit that developing such an evaluation matrix will require much more thought than we were able to give to it. We attempted to do so for a DoD kind of environment, and were not entirely successful. The major complication is the interaction between the environments in which the "good guy" and the "bad guy" are supposed to operate. A further complication is assessing how effective personnel security measures are: in an environment containing highly sensitive information, even a trusted person is subject to temptations beyond anticipation or easy control. In particular, even though the good guy might decline to directly access sensitive information on behalf of another, he might be induced to plant system modifications to permit the bad guy to have access. And there is always the chance that an accidental exposure to highly sensitive (=valuable) information will tempt someone to exploit his knowledge of that information, even though he would never have intended to deliberately acquire the information.

The values filled in the matrix would initially be lists of the necessary features (from the hierarchy), probably in the form of an eight-entry vector.

5.3 Approved Products List

To help the procurement process, the next step would then be to agree upon a small number of typical environments (e.g., commonly occurring intersections in the matrix) and to publish a list of those products that meet or exceed the requirements for each such environment. The purchaser of a system would then only have to characterize his system as, for instance, being of Type 1A, and then he would have a list of approved products to buy (in so far as security is concerned.)

Although the process described here has not yet been formally adopted, technical experts within the DoD Computer Security Consortium are sufficiently familiar with the relevant issues and confident enough in the technology to have drawn up a draft list of six major categories into which DoD systems fall, with each category characterized by the major features from the evaluation metric that must be present.

Although this list is not directly applicable to non-DoD environments, it is nevertheless illustrative. The list contains four components:

- o The category number.
- o The allowed kind of applications.
- o The kind of multi-level security allowed (mix of classification levels). These are abbreviated as follows: TS = TOP SECRET, S = SECRET, C = CONFIDENTIAL, U = UNCLASSIFIED.

o The major features from the security metric that must be present.

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

5.4 Administrative Aspects

We are mindful that the administrative aspects of the proposed process are formidable. Rough analogies can be found in current procurement practices. It should be noted that the whole process is somewhat akin to what is being done for pollution control. Public policy vaguely establishes a desire for cleaner air, whereas the technical discussions between the government and the automobile manufacturers establish approximately what is feasible. The policy setters, however, set the acceptable levels somewhat higher than industry would wish, and furthermore announce a general plan of tightening the standards over time. The same kind of process is envisioned here. The evaluation matrix is in effect the security standard analogous to the pollution level standard. At first, the entries in it would be fairly low, but as technology matures (mostly the assurance technology) the standards would be tightened. Publication of intention to follow this process, along with the timetable of standards-tightening, would encourage vendor development, provided it is all done in a reasonable, public, and well-informed manner.

6. SPECIAL SOLUTIONS

Note: Special thanks and credit to Clark Weissman for most of the material in this section.

The above recommendations are phrased in the context of a single operating system that is supposed to securely support multiple users of differing degrees of trustworthiness simultaneously handling data of differing degrees of sensitivity, i.e., what has become to be known as the classical computer security problem. As we have observed, true solutions to this particular problem may well not be widely available for a decade. By sacrificing some of the operational generality or efficiency of having a truly secure multi-use system, government agencies can achieve (and have achieved) secure processing of sensitive data, while still retaining some amount of resource sharing. (The security problems we have been addressing here all arise because of the desire to share some resource -- be it hardware, operating system software, or data itself -- between users of drastically different security profiles.) We feel it important to draw our readers' attention to the major techniques involved, even though strictly speaking they are outside our scope since they are ways of "avoiding" the operating system security problem rather than of solving it. The techniques discussed here are all (or will shortly be) standard practice for DoD and similar applications and are generally equally applicable to other environments.

We recognize the many deficiencies -- both regarding security effectiveness and operational efficiency -- in the approaches discussed here and must admit to considerable internal debate within our committee about them. These solutions do, however, provide a means for the operators of current systems to gradually evolve toward a recognition of their security problems and toward a partial solution of them. We suggest these special solutions be considered in any serious long-range plans for security improvements of existing products or installations.

6.1 Periods Processing

Periods processing is the term used for processing a given type of sensitive information on a computer, exclusively, for a limited period of time. All other applications and data uses are prohibited during that given time period. Processing of other sensitive data requires a shutdown of the computer, and a careful sanitization of all memory, storage, and printer devices. This transition is called a "color change." It is labor intensive, slow (on the order of thirty to sixty minutes) to effect, breaks operational continuity, and often under-utilizes computer resources, since the machine cannot be shared. It also cannot be used where the nature of the application requires simultaneous access to the same information by people of different security profiles. (This deficiency is generally true of all the special solutions discussed here, except for the secure-subsystem one.) However, it is current practice, offers little security or technical risk, and has no run-time overhead.

6.2 Automated Periods Processing

The objective of this solution is to reduce the manual efforts and attendant time lost in color change between processing of data of different sensitivities. Two approaches have been designed for effecting this automatic switch over: the Job Stream Separator (JSS) [18] and the Crypto Switch. Both schemes involve the use of an auxiliary mini-computer to effect and control the color-change operation. The auxiliary computer is a shared mechanism between the two time periods and, hence, requires considerable trustworthiness. It is a simpler mechanism than the larger computer it controls and is within the technological state of the art. DoD development of these approaches is currently in progress.

6.3 Secure Distributed Processing

The architectural strategy of this solution is to use a computer network to tie together a collection of computers, each of which is dedicated to the processing of a different level or kind of sensitive data. Users could then view the network as a multi-level, or mixed-

data sensitive "supra-computer." Access control functions within the network restrict users to operation on only those dedicated computers to which they have authorized access rights. The fundamental technology to make secure distributed processing possible is that of End-to-End Encryption. End-to-end encryption guarantees that message text remains enciphered regardless of the communications path from the originator to recipient, e.g., from user terminal to host computer. Note that it in principle eliminates the need for a secure network itself, although extreme care must be taken. With the adoption by the National Bureau of Standards of the Data Encryption Standard (NBS-DES) [13], major strides toward realizing end-to-end encryption technology have been made and it is close at hand. Operational studies have shown that end-to-end encryption technology is a cost-effective technology for safe transmission of sensitive data (with the NBS-DES being cryptographically strong enough for all non-national security data, and possibly even for some of that within some constraints), for access control to dedicated computer resources, and for authentication of users and computer hosts. Various trusted devices and processors are needed to make secure distributed processing a reality. These devices include smart encryption boxes, secure terminal handlers, and secure network front ends. Many of the problems encountered in a general-purpose solution also arise, although in a possibly less severe form, in this kind of solution.

6.4 Secure Subsystems

Given an untrusted operating system, this approach employs the use of a trusted transaction data management system or other trusted special-purpose subsystem in concert with facility and procedural constraints that limit the population of users to the trusted subsystem. (Only trusted users are allowed access to any parts of the system outside of the trusted subsystem.) This solution combines trusted software (but not the operating system itself) and trusted procedures, and is an expedient until completely trusted operating systems are more widely available. Secure subsystems development for the DoD in limited transaction applications is currently under way.

6.5 Assurance of Special Solutions

As observed, each of the above special solutions (except periods processing, section 6.1) requires a certain amount of "trusted" software. This software must therefore be subjected to a technical evaluation that determines whether such trust is warranted for a particular application. The criteria espoused above in section 5.1 are generally applicable and a process similar to that discussed in section 5 must be followed. Since most of the software is encapsulated in an environment that isolates it from user programming, the extreme degree of confidence required for the general solution (a secure general-purpose operating system) is not required, and therein lies the promise for ready adoption of such techniques. The major difficulties lie in the secure subsystems approach, since it is essential that "untrusted" users be locked into the secure subsystem so that they may not exploit the underlying insecure operating system's weaknesses.

7. REFERENCES

Note: No attempt has been made to make this even a representative bibliography. For further material, see Carlstedt [5], almost any of the other references, or any of several more comprehensive, although not necessarily current, bibliographies.

- [1] Abbott, R.P., et al, Security Analysis and Enhancements of Computer Operating Systems, The RISOS Project, Lawrence Livermore Laboratory, Livermore, Ca., National Bureau of Standards, Washington, D.C., NBSIR 76-1041, April, 1976.
- [2] Anderson, James P., Computer Security Technology Planning Study, James P. Anderson and Co., Fort Washington, Pa., USAF Electronic Systems Division, Hanscom AFB, Ma., ESD-TR-73-51, vols. I and II, October, 1972. (AD 758206 and AD 772806)
- [3] Attansio, C. R., Markstein, P. W., and Phillips, R. J., "Penetrating an Operating System: A Study of VM/370 Integrity," IBM Systems Journal, Vol. 15, No. 1, 1976, pp. 102-116.
- [4] Bisbey, Richard, II, Popek, Gerald, and Carlstedt, Jim, Protection Errors in Operating Systems: Inconsistency of a Single Data Value over Time, USC Information Sciences Institute ISI/SR-75-4, Dec. 1975.
- [5] Carlstedt, Jim, Protection Errors in Operating Systems: A Selected Annotated Bibliography and Index to Terminology, USC Information Sciences Institute, Marina Del Rey, Ca., ISI/SR-78-1, Feb. 1978. (AD 053016)
- [6] Gold, B.D., et al, "VM/370 Security Retrofit Program," Proceedings of the National ACM Conference, October, 1977, pp. 411-418.
- [7] Jones, Anita K., and Wulf, William A., "Towards the Design of Secure Systems," Software -- Practice and Experience, vol. 5, no. 4, Oct.-Dec., 1975, pp. 321-336.
- [8] Karger, P. A., and Schell, R. R., Multics Security Evaluation: Vulnerability Analysis, USAF Electronic Systems Division, Hanscom AFB, Ma., ESD-TR-74-193, Vol. II, June 1974. (AD A001120)
- [9] Lampson, Butler W., "A Note on the Confinement Problem," CACM, vol. 16, no. 10, Oct. 1973, pp. 613-615.
- [10] Linde, R. R., "Operating System Penetration," Proceedings of AFIPS 1975 National Computer Conference, Vol. 44, pp. 361-368.
- [11] Linden, T. A., "Operating System Structures to Support Security and Reliable Software," ACM Computing Surveys, vol. 8, no. 4, Dec. 1976, pp. 409-445.
- [12] Lipner, S. B., SATIN Computer Security, The MITRE Corp., Information Systems Technology Applications Office, USAF Electronic Systems Division, Hanscom AFB, Ma., MCI-75-2, Sept. 1972.
- [13] National Bureau of Standards, Data Encryption Standard, Federal Information Processing Standards Publication 46, Jan. 1977.
- [14] Neumann, P. G., "Computer System Security Evaluation," Proceedings of AFIPS 1977 National Computer Conference, Vol. 46, p. 1087-1095.
- [15] Neumann, P. G., et al, A Provably Secure Operating System: The System, Its Applications, and Proofs, Final Report, Project 4332, SRI International, Menlo Park, Ca., 11 Feb. 1977.

- [16] Rzepka, W. E., Considerations in the Design of a Secure Data Base Management System, RADC-TR-77-9, Mar. 1977. (AD A039169)
- [17] Saltzer, J. H., and Schroeder, M. D., "The Protection of Information in Computer Systems," Proceedings of the IEEE, Vol. 63, No. 9, September, 1975, pp. 1278-1308.
- [18] Schacht, J. M., Jobstream Separator: Supportive Information, The MITRE Corp., USAF Electronic Systems Division, Hanscom AFB, Ma., ESD-TR-75-354, Jan. 1976. (AD A020521)
- [19] Schaefer, M., and Hinke, Thomas H., Secure Data Management System, Systems Development Corporation, TM-(L)-5407/007/00, RADC-TR-75-266, Nov. 1975. (AD A019201)
- [20] Ware, W. et al, Security Controls for Computer Systems, RAND Corp. Technical Report R-609, 1970.
- [21] Secure Minicomputer Operating System (KSOS), Computer Program Development Specification (Type B-5), Department of Defense Kernelized Secure Operating System., Ford Aerospace and Communications Corp., WDL-7932, Sept. 1978.
- [22] Final Report, PROJECT GUARDIAN, Honeywell Information Systems, Inc., Federal Systems Division, ESD-TR-78-115, Sept. 1977.
- [23] Weissman, Clark, "System Security Analysis/Certification," notes for talk presented at the MITRE Computer Security Workshop, Jan. 31, 1979.
- [24] Popek, G. J., et al, "UCLA Secure Unix," Proceedings of the 1979 National Computer Conference, June, 1979, pp. 563-572. [to appear]
- [25] "Challenges of Protecting Personal Information in an Expanding Federal Computer Network Environment," report of the Comptroller General to the Congress of the United States, LCD-76-102, April 28, 1978.

While these proceedings were in preparation, the following report amplifying, and slightly modifying, the proposed evaluation approach of section 5 has been written as part of the activities of the DoD Computer Security Initiative:

- [26] Nibaldi, G.H., Proposed Technical Evaluation Criteria for Trusted Computer Systems, MITRE Corporation, Bedford, Mass., 25 October 1979, MITRE Report #M79-225.

PART IX: SESSION 7

APPLICATIONS AND NON-INTEGRATED DATA FILES

Chairperson: Gerald E. Short
TRW Systems

Participants:

Robert P. Abbott
EDP Audit Controls

Robert S. Roussey
Arthur Andersen & Co.

Walter L. Anderson
U.S. General Accounting Office

Robert Stone
Uniroyal Corporation

Sheila Brand, Recorder
Dept. of Health, Educ., & Welfare



From left to right: Robert Stone, Walter L. Anderson, Gerald E. Short, Robert S. Roussey, Robert P. Abbott, Sheila Brand.

Note: Titles and addresses of attendees can be found in Appendix B.

EDITOR'S NOTES

G. E. SHORT

For the last seven years, Mr. Short has been involved in the performance and management of data security studies. These studies have included penetration studies, security kernel technology, packet-switched networks, formal verification, and data base management development. He has been a member of the NBS sponsored task force on data security and privacy (TG15), panel participant on Data Security, and a member of the CBEMA committee on Data Encryption. He has presented papers on computer security at GUIDE, SHARE, NCC, and ARMY and NSA sponsored symposiums.

SHEILA BRAND

Sheila Brand is the Senior Adviser for Computer Technology in the Office of the Inspector General (OIG), U.S. Department of Health, Education, and Welfare. In this capacity she acts as a consultant to auditors and investigators in areas of computer systems security, computer crime, and computer auditing as well as in the management of the system development process. She comes to OIG from the Social Security Administration where she initiated their security program; acted as project and task force leader for a number of large risk analyses efforts; and served as the SSA representative to the NBS Public Advisory Group on Computer Security TG/15 where she chaired the Internal Controls working group. Before joining the Federal government she was a senior systems analyst with Commercial Credit Corp. Her assignments there included operating system maintenance, and project leader for design of DBMS and operating system security for a nationwide real-time transaction system. She holds a B.A. in Mathematics with a Minor in astronomy.

THE CHARGE TO THE GROUP

This session was to address the vulnerabilities and necessary controls related to applications, application program development and maintenance, and data files where a DBMS is not employed. [See PART I, Section 2 for the complete charge given to this group.]

This session had great difficulty in coming to a consensus position. Also the Chairperson was unable to complete a report for the group. Ms. Brand therefore undertook to present a paper that reviews two current approaches, describes the consensus view that seemed to be emerging from the session, and discusses their common threads. The paper was circulated among the attendees for concurrence and therefore is a consensus report.

AN APPROACH TO IDENTIFICATION AND AUDIT OF VULNERABILITIES AND CONTROLS IN APPLICATION SYSTEMS

Sheila Brand

1. INTRODUCTION

The charge of this session was to identify vulnerabilities and specific controls which when applied would deter and/or detect exploitation of vulnerabilities associated with:

- * applications
- * application program development
- * application program maintenance
- * non-DBMS data files

In addition, we were asked to provide qualitative effectiveness measures of identified controls as to their ability to increase the difficulty of exploitation of specific vulnerabilities. Given the limited time frame of the workshop, and the basic workshop assumptions (Section 2.1) the participants agreed that this last assignment would not be attempted.

1.1 Complexity of Problem

Our job proved to be extremely difficult. Not only was the time available a constraint, but our actual mandate as described in Section 1.0 proved most frustrating. Two basic problems were quickly identified by the participants:

- * The task of defining all vulnerabilities inherent in applications and assigning specific controls was not feasible.
- * Limiting the scope of consideration rigorously to the application area alone did not seem satisfactory. The interaction of applications with other parts of the system and organization call for a more comprehensive treatment.

Because of these difficulties, it was not possible to complete our task within the three days of the workshop. To augment the conclusions from our activities and in an attempt to satisfy NBS's needs, the author undertook a review of available work in application systems security. This report is a combination of the results of this review and conclusions reached by the session participants. Grateful acknowledgement is given to Robert Abbott for his contribution of Appendix B which provides a discussion of fundamental security concepts in the writing of computer programs; to Walter Anderson for his help in summarizing the session deliberations and in providing Figure 4, a summary chart of controls; and to Robert Roussey for his elaboration of the Arthur Andersen and Company approach to Auditing of Computer Systems.

The conclusion of this work is that providing definitive lists of controls for the deterrence of vulnerabilities can be approached from many angles -- all of which are useful.

For example, in the first strategy, the matrix approach, one can take a highly structured approach by developing detailed lists of controls which will protect against known vulnerabilities inherent in components of a data processing system. This methodology lends itself to utilization of checklists or matrices to show the cross-relationship between safeguards and exposures vs. assets to be protected.

A second strategy, the NBS approach, makes use of the concept of system control objectives and partitioning the problems according to phases of the application system life cycle. Using the life cycle one comes closest to providing insight into vulnerabilities and controls associated with development and maintenance phases of the application system life cycle -- two workshop charges given our session.

A third strategy, a hybrid approach, which also makes use of the concept of system control objectives, partitions the problem by taking a transaction flow approach. Here controls are applied as a result of "overlaying" control objectives onto each component of this flow.

1.2 Scope of Report

The remainder of this paper will provide descriptions of the three approaches which use the strategies outlined above. Section 2.0 will give some basic definitions of terms and assumptions used throughout this report. Section 3.0 presents the matrix approach, Section 4.0 the NBS strategy, and Section 5.0 the outline of a hybrid strategy developed by the session's participants. Section 6.0 provides some conclusions and recommendations.

Section 3.0 comes closest to fulfilling the workshop charge of listing vulnerabilities and controls for the application. It is derived from procedures developed by Dr. Jerry FitzGerald and is detailed in his book, Internal Controls for Computerized Systems [1]. This approach focuses on a matrix containing controls which interrelates an organization's vulnerabilities (concerns/exposures) with specific resource/assets that must be protected within the domain of the program/computer processing environment.

Section 4.0 presents an approach developed by the NBS with assistance of the now defunct NBS Public Advisory Group TG/15, entitled Computer Systems Security. This strategy emphasizes placement and making use of controls in the application at all stages of the system life cycle. NBS stresses the need for identifying security objectives for the protection of data; assessment of data sensitivity; and the vulnerabilities inherent in a specific system design. No attempt is made to provide checklists of controls vs. vulnerabilities though many of each are discussed. However, the issues associated with application system design, development, and maintenance are best addressed by this approach. The NBS approach will soon appear in their guideline publication, Security for Computer Applications [2].

Section 5.0 describes an approach developed by the workshop participants. It combines the concept of system control objectives as detailed by Arthur Andersen and Company[3] with controls delineated by SRI in their Systems Auditability and Controls Study[4]. This approach falls midway between the rigid matrix method and the general NBS method.

2. DEFINITIONS AND ASSUMPTIONS

Principal terms used throughout this paper are defined as follows:

Application system. A set of logically related computer programs and associated manual activities designed to accomplish specific objectives or functions. The application system runs under the direction of the operating system and depends on it for basic security protections.

Sensitive application. An application system requiring a degree of protection because it processes sensitive data or because improper operation or manipulation would result in significant loss or harm.

Sensitive data. Data requiring a degree of protection due to the risk and magnitude of loss or harm which could result from inadvertent or deliberate disclosure, alteration or destruction.

Data integrity. The state that exists when computerized data is the same as that in the source documents or has been correctly computed from source data and has not been exposed to accidental or malicious alteration or destruction. Erroneous source data and fictitious additions to the data are also considered violations of data integrity.

Data confidentiality. The state that exists when data is held in confidence and is protected from unauthorized disclosure. Misuse of data--by those authorized to use it for limited purposes--is also a violation of data confidentiality.

ADP availability. The state that exists when required ADP services can be obtained within an acceptable period of time.

Application system life cycle. The life cycle of a computer application consists of three identifiable phases--initiation, development, and operation. After some period of operation, the system will have to undergo an expansion or revision, and the life cycle is then repeated.

Initiation phase. The initiation phase establishes the objectives and general requirements of the computer application. System planners consider alternative approaches for a target system. Based upon feasibility studies and cost-benefit analyses of the potential solutions, a decision to proceed with the development of a specific system is reached.

Development phase. The development phase consists of four stages--definition, design, programming, and testing. While these are logically independent stages, in practice they may overlap substantially.

Operational phase. The operational phase begins once the system has been accepted by its intended users and they become dependent on it to fulfill their organization's mission and responsibilities.

2.1 Overall Workshop Assumptions

In order to limit the scope of work of each session, NBS provided the following two assumptions:

- * The application system functions in a multi-user teleprocessing environment.
- * System vulnerabilities are to be identified, but their probability of occurrence was to be ignored. In other words, risk analysis considerations were not to be included in the deliberations.

2.2 Session Assumptions

Session participants made the following additional assumptions:

- * Policies and guidelines exist which define adequate levels of protection for physical, personnel, and communications resources associated with the environment of the application.
- * Policies and guidelines exist defining acceptable operating modes for computer application.
- * There are limitations in providing definitive measures for protective application system controls. This is due to the wide variety of application systems and the shortcomings of current and near term technology.
- * No attempt would be made to address the problem of data sensitivity. Though OMB Circular A-71, Transmittal No. 1 talks in terms of sensitive applications and sensitive data, no Federal guidelines have yet been provided for the classification of non-national security-type data into national-security-type categories.

3. THE MATRIX APPROACH

This section summarizes the strategy for evaluating application system vulnerabilities and controls which appears in Dr. FitzGerald's book, Internal Control for Computerized Systems [1]. It is the most rigorous of the three methodologies described in this report and probably comes closest to satisfying the Workshop request for identifying controls and vulnerabilities of the application.

The matrix approach subdivides the data processing function into nine components. Controls relating to each component are then enumerated. FitzGerald has identified the following nine components:

- * General Organization
- * Input
- * Data Communication
- * Program/Computer Processing
- * Output
- * On-line Terminal/Distributed Systems
- * Physical Security
- * Data Base
- * System Software

For each component a matrix is developed which identifies specific controls which address an organization's concerns/ exposures for protection of the resources/assets peculiar to that component of the overall system.

As our session was concerned with the application system, it is useful to focus on the specific matrix for that component of the overall processing system which deals with Program/Computer Processing.

3.1 Application System Resources/Assets

Application system resources and assets to be protected have been identified by FitzGerald as [see [1], pages 36,38]:

- " * Application Programs and Systems. Any or all the computer programs that are utilized in the data processing operations. This resource should also be viewed as the overall macrosystems that operate within the organization (these systems may be made up of a group of computer programs). This is far and away the most valuable asset of the organization because, in the long run, the computer programs are more costly than the hardware upon which they operate.
- * Data Record Integrity. The data that is stored in the computer files or data bases and is used in the everyday processing of the organization's computerized record-keeping system.
- * Output Integrity. The believability and integrity of the output reports from the system. The auditor should review this resource to insure that the output reports are Consistent, Accurate, Timely, Economic, and Relevant to the intended purpose (reports that meet these criteria will CATER to the needs of the organization).
- * Central System. Most prevalent in the form of a central computer in which the computer programs operate. This asset may be in the form of a central computer system, or it may be in the form of numerous computer systems spread around in a distributed network.
- * Software Programs. The software programs that run the overall computerized systems. These may include the operating system software (usually supplied by the computer vendor) as well as the software programs utilized to maintain and operate the data communication network, or the data base system (data management software). These software programs usually operate at the "systems control level" because any controls that are built into, or programmed into, this level of software affect all application programs. For example, a control that is built into the operating system software, data communication control software, or data management software would have its effect upon any incoming transaction that passed through that level of software programs without regard to whether it was a payroll transaction, inventory control transaction, financial balancing transaction, or the like."

3.2 Application System Concerns/Exposures

FitzGerald has identified the following concerns/exposures [see [1], pages 35-36]:

- " * Program Errors and Omissions. The accidental or intentional creation of an error during the processing of the data or the running of the application programs, including the accidental or intentional omission of data (loss) during the processing of a computer program. This type of exposure includes, but is not limited to, multiprogram code, trapped machine checks where programs just quit processing, loss of data during the running of a program, and the like.

- * Unauthorized Program Changes. The temporary or permanent change of program code by individuals who are unauthorized to make these changes, as well as by individuals who are so authorized but who make illegal program changes for whatever reason.
- * Security/Theft. The security or theft of information or programs that should have been kept confidential because of their proprietary nature. In a way, this is a form of privacy, but the information removed from the organization does not specifically pertain to an individual. The information or computer programs might be inadvertently (accidentally) removed from the organization or might be the subject of outright theft.
- * Data Validation. The computer program editing of data prior to its processing and the preprogrammed specific actions that should be taken when erroneous data is discovered (this may also include the discovery of omissions in certain data that should have been included).
- * Hardware Errors. The malfunctioning of the computer hardware so it appears that a program has made some sort of an error in processing. The concern here is that a hardware malfunction may cause erroneous data, data omissions, loss of specific data, and the like.
- * Restart and Recovery. The restarting of computer programs that have failed during their normal course of processing and the recovery that should take place so no data is lost, erroneously processed, or processed twice because of the failure (the failure may have been caused by program failure or computer hardware failures).
- * Audit Trails. Insurance that the processing of the data can be traced backward and forward through the entire computer processing cycle.
- * Computer Program Generated Transactions. Insurance that any transactions that are automatically generated within an on-line system are adequately controlled. In other words, some on-line systems automatically create transactions during the time they are being run and these transactions should have adequate controls to prevent errors, erroneous transactions, and illegal transactions.
- * Error Handling. The procedures and methods used to insure that all transactions or data that are rejected during the computer processing are, in fact, corrected and reentered into the system in a timely manner. This involves accounting for and detecting data errors, loss, or the nonprocessing of transactions, as well as the reporting of these errors, error correction, and the corrected data resubmission."

3.3 Application System Controls

Figure 1 shows FitzGerald's Program/Computer Processing Control Matrix. Across the top are columns for the organization's concerns/exposures (vulnerabilities). Down the left side are rows for the resources/assets to be protected. Numbers in the squares indicate detailed controls which will protect a specific resource/asset from a specific concern/exposure. To illustrate the safeguards enumerated by FitzGerald, controls numbered 1 through 10 (out of the 91 controls in Figure 1) are listed below [see [1], pages 38-40].

- "1. Transactions that are consecutively numbered by the station transmitting (these might be computer generated transactions) to the computer should be sequence number checked by the computer programs. In other words, the computer programs should verify the unbroken sequence of input or output transactions and take corrective action, should there be a break in sequence. One form of corrective action would be to notify the terminal operator and to

		CONCERNS/EXPOSURES								
RESOURCES/ASSETS		PROGRAM ERRORS AND OMISSIONS	UNAUTHOR- IZED PROGRAM CHANGES	SECURITY/ THEFT	DATA VALIDATION	HARDWARE ERRORS	RESTART AND RECOVERY	AUDIT TRAILS	COMPUTER PROGRAM GENERATED TRANS— ACTIONS	ERROR HANDLING
	APPLICATION PROGRAMS AND SYSTEMS	1-9, 16-20, 63, 71-74, 86, 89, 90	13, 22, 23, 25, 30-33, 35, 38, 43, 44, 49-51, 70, 85, 88, 90, 92	24, 25, 30-35, 38, 43, 49, 62, 70	2-8, 16, 69, 72-74, 89	70, 72-74, 89	20, 76, 77	6, 16, 22, 23, 25-28, 30, 32, 33, 35, 43, 49	1, 5, 7, 43, 44, 51, 64, 65, 67, 68, 70, 71, 90,	9-15, 45, 71 78-81, 83
	DATA RECORD INTEGRITY	1, 3-7, 16, 17, 27, 69, 71, 73, 74, 80, 81	30-33	24-26, 29, 30-35, 38, 39, 48, 49	1-5, 7, 10, 12-14, 16, 17, 24, 25, 27, 28, 63, 69	41, 46, 47, 72, 87	20, 21, 28	6, 16, 25-28, 31, 32, 49, 64 66, 80-82	64-68	10-15, 45, 79-83
	OUTPUT INTEGRITY	2-4, 6, 7, 16, 18-20, 27, 69, 71, 73, 74, 80, 81	39	24-26, 29-35, 38, 39, 48, 49	2, 4-7, 9, 10, 16, 28, 69		21, 28	17	64, 65, 67, 81	15, 71, 75, 78-83
	CENTRAL SYSTEM	16, 40-42, 46, 47, 69, 76, 77, 91	36-39, 48, 49, 51-53, 85, 86, 91	36-39, 42, 47-49, 51-53, 55, 62, 84, 91	16, 40, 69	46, 54-61, 87	42, 76, 77	30, 33, 42, 84, 91		91
	SOFTWARE PROGRAMS	1, 86, 87, 90	13, 22, 23, 25 30-33, 35, 38 44, 49, 50, 51, 70 85, 88-90, 92	24, 25, 30-35, 38, 49, 62, 70	89	70, 87, 89			51, 70, 90	

FIGURE 1 PROGRAM/COMPUTER PROCESSING CONTROL MATRIX (FROM FIG. 5-1, [1])

close down the transmitting station's ability to transmit data until the remote station takes some sort of corrective action.

2. Have the programs compare the total count of input transactions to a predetermined total count or to a count of output transactions.
3. Let the program perform automated and/or preprogrammed editing for all input after it gets into the computer. Some of the editing that the program can perform might be as follows:
 - Count the number of fields in a record and compare that with a predetermined number of fields.
 - Check for the reasonableness of the input data with regard to some set of preestablished boundaries.
 - Test the data for blanks, sign (plus or minus), numeric, or alphabetic, and compare that with preestablished criteria.
 - Check for consistency between fields of an input transaction (this would be a specific control with regard to a specific application input).
 - Conduct a limit test, and reject data or take corrective action whenever the data falls outside of some limit or predetermined range.
 - Check for completeness of data, for example, the zip code field should be full, and it should contain numeric data only.
 - Conduct sequence checking in order to insure correct sequence.
 - Conduct data checking in order to insure that the dates are correct whenever this is applicable.
 - Use self-checking numbers that pinpoint erroneous entry of account numbers or whatever type of number the organization is using.
 - Enter critical data twice on one transaction input and have the computer programs cross-check these two inputs to insure that, first it was entered correctly, and second there was no error during transmission.
4. Let the computer programs compare or crossfoot predetermined control figures such as:
 - Record counts
 - Control totals
 - Hash totals
 - Batch control totals
5. Have the program recompute various totals of significant financial or accounting figures and transmit these totals back to the original input station.
6. Have the programs prepare specific reports that will display the contents of batch controls, header controls, and any other types of control totals that can be sent back to the original station that inputted the data.
7. Have the programs compare the current data totals with historical totals in order to maintain a logical relationship over time.

8. Have the programs perform logical relationship tests. Logical relationship tests are solely dependent on a specific application because there may be logical relationships within a specific application system.

9. Have the programs look for duplicate entries of data. Whenever duplicate entries are suspected, the original station inputting the data should be immediately notified.

10. Design systems so upon the discovery of erroneous data during processing, the original entry station is immediately notified so correction can take place as soon as possible."

3.4 Limitations of the Matrix Approach

The matrix approach gives the user (auditor, designer, security officer, etc.) the ability to quickly review relevant controls for the protection of a general asset from a general vulnerability; or it allows the user to start with a general concern and review controls which would act as deterrents to this concern. It is still a user task to narrowly define these variables for the application under consideration. In other words, go from the general to the specific.

The matrix methodology may tend to oversimplify the problem definition. By introduction of a checklist type approach a user may ignore many problems interrelated to, but apart from the application system. It would be necessary to use all nine matrices to get a well-rounded view of all the vulnerabilities against which an application system must be protected. And, even if this extremely cumbersome and complicated process is undertaken (FitzGerald's book contains over 650 controls in nine matrices) there is still a good possibility that the user's environment will not be completely described. No checklist can cover all potential applications, i.e., be all inclusive.

In addition to the "volume" problem, this approach does not address redundancy or sharing of protective measures. In a secure system one control may serve to protect against more than one vulnerability or the same vulnerability within a series of separate applications sharing common resources. An example of such a control would be an authorization/identification table imbedded in the telecommunication front-end software which serves as an access control for a number of applications in a multi-user/multi-purpose on-line system. It would be useful if the capability for assessing the effectiveness of these blanket-type protective devices were available within the matrix methodology.

One last comment on the checklist approach. By using this technique, audit findings tend to yield "yes/no" answers. Either a control is there or it isn't. The question still remains as to who is responsible for qualifying the final results. Does the auditor have the responsibility for final translation of the yes/no list into a qualitative analysis of the security's effectiveness? Or, does the auditor simply relate findings and leave it up to management to do the qualitative assessment?

4. THE NBS APPROACH

NBS has developed a general strategy which is presented in its draft document: Security For Computer Applications [2]. This guideline classifies undesirable computer events in terms of their general effects on computerized data rather than in terms of their ultimate effect such as denial of benefits, or loss of money or resources.

4.1 Vulnerabilities and Security Control Objectives

The NBS classification of undesirable events (vulnerabilities that are activated) relates them directly to three general security control objectives for all application systems. The vulnerabilities and their countering control objectives are:

- * Modification or destruction of data -- Data Integrity.
- * Disclosure of data -- Data Confidentiality.
- * Unavailability of data or system service -- ADP Availability.

All controls fall into categories whereby they meet one or more of these control objectives. In assessing the security needs of an application, one would first identify major objectives and then choose specific controls which would meet these objectives.

4.1.1 An Example.

As an example, if one were designing a payroll system, the security "reasoning" would go something like this:

The major causes of loss in a payroll system are errors. The system must also be protected against fraud, embezzlement, and theft. Therefore, the primary concern would be for the prevention of errors -- the corresponding control objective being data integrity. Data integrity would also cover the problems related to fraud, embezzlement and theft.

Payroll systems frequently involve sensitive, personal, or other confidential data. Therefore, a second control objective would be data confidentiality.

The extent to which a payroll system requires ADP availability is a function of the ability of the organization to fall back on manual procedures and the cycle for disbursements.

Using this approach, the system planner would then place primary emphasis on design of safeguards to ensure data integrity, with lesser emphasis on confidentiality and availability.

4.1.2 Types of Vulnerabilities.

NBS has not coupled vulnerabilities with specific controls. However, they have compiled a lengthy list of vulnerabilities [5] which occur in an application environment. Appendix A contains the complete list. Areas covered include:

- * Erroneous or Falsified Data Input
- * Misuse by Authorized End Users
- * Uncontrolled System Access
- * Ineffective Security Practices for the Application
- * Procedural Errors Within the ADP Facility
- * Program Errors
- * Operating System Flaws
- * Communications System Failure

4.1.3 Six Control Categories.

As to controls, NBS has provided detailed discussion of six basic control categories and indicated the general problems that each will address. Included are:

- * Data Validation
 - Consistency and reasonableness checks
 - Data entry validation
 - Validation during processing
 - Data element dictionary/directory
- * User Identity Verification
- * Authorization
- * Journalling
- * Variance Detection
- * Encryption

4.2 The System Life Cycle

A second emphasis within the NBS approach is placement and use of appropriate controls at each stage of the system life cycle. The life cycle consists of three phases: initiation, development, and operation. Of help in fulfilling the workshop charge are steps recommended by NBS in its discussion of the first two of these phases: initiation and development.

The NBS discussions related to the initiation and development phase provide excellent guidelines and insight into problems, vulnerabilities, and controls which should be used during application system development and maintenance. A summary of key points follows.

4.2.1 The Initiation Phase.

During this phase overall system requirements, objectives and sensitivity are defined. Basic security feasibility analysis should be performed to assure that the application design allows for the building in of cost-effective security. The designers should be able to give affirmative answers to the following questions:

- * Will the source data supplied to the ADP system be accurate and sufficiently complete to support its intended users without harmful side-effects?
- * Can users of the system be adequately identified and authenticated so that they can be held accountable for their actions?
- * Are user interfaces to the system sufficiently restricted so that adequate security is feasible?
- * Do the boundaries between ADP and related manual activities provide maximum separation of duties and independent review?
- * Is the proposed processing facility adequately secure?

- * Have the impact and frequency of major security failures been taken into account in the design? Specifically, have the following been accounted for: inaccurate data, falsified data, disclosed data, lost data, or unavailability of data or services?

4.2.2 The Development Phase.

The development phase includes the activities of security requirements definition, design, programming and testing. Some actions that will help assure an effective security system for the application should be taken during the development phase.

In the definition stage the designer should:

- * Define security requirements carefully making sure that the plan specifies which vulnerabilities are to be controlled by software, which by hardware and which by administrative means.
- * Identify each job function related to the application and how it interfaces with the system, supplies data to the system, or supports the system. For each of these functions identify organizations and their responsibilities. Identify the controls and vulnerabilities associated with each group.
- * Analyze each function for separation of duties.
- * Identify all data associated with the application whether it be input, output or stored data. Assess the sensitivity of the data. Define security requirements for protection of the data with respect to the objectives of data integrity, confidentiality and availability.
- * Define disaster and error recovery plans.
- * Specify security requirements concisely--do not hedge.

In the design stage the planners should:

- * Design controls that are easy to use employing the concepts of human engineering.
- * Restrict terminal access capabilities to minimum user requirements. (Example - transaction systems which do not provide on-line programming capability are more protected from tampering than time-sharing systems that permit programming.)
- * Perform a design review to identify weak points in the security plan. (This should be done by an independent group after the security scenario is complete but before programming has begun.)

During the programming stage, the following steps should be taken:

- * Protect application system code and data by running development activities on a separate computer or at a time period when live applications are not in execution.
- * Employ peer review to assure that code does not contain trap doors, trojan horses, or other security errors and that it satisfies all design criteria, is efficient, easily maintainable and well documented.
- * Use a program library that can: restrict access to program modules to only authorized persons; record all access and modifications; associate record and byte counts with program modules to help detect changes in a module.

- * Maintain complete and current documentation for all security software.
- * Use high level languages which support structural control flow, extensive data definitions facilities, type checking, and well-defined module calling definitions.
- * Establish a control objective to eliminate coding structures which have an adverse effect on application system security and integrity. [This bullet was not taken from the NBS applications guidelines document. It was suggested by R. Abbott and is elaborated on in Appendix B.]

Test stage steps include:

- * Stringent testing of security controls is necessary to assure that they are reliable, meet specifications, and meet user requirements. The tests should include runs which demonstrate how the controls respond to normal, abnormal, unusual, improbable, and illegal circumstances.
- * Static tests which employ code review and penetration studies of the system documentation and code may be the most effective way to detect trap doors and other unauthorized codes. However, in a large system the job may prove too complex to be done effectively.

4.3 Limitation of this Approach

The NBS strategy provides insight and excellent general approaches to initial application design and is also very useful for making major system modifications. However, it is not, nor was it ever meant to be, a road map for performing speedy system reviews or audits.

To develop guidelines which can be used for both design and audit, a combination of the matrix and NBS methods would be best. The strategy which this session's participants started to develop appears to contain the elements of the necessary merger.

5. THE SESSION APPROACH

As mentioned earlier the group found our mandate extremely difficult, if not impossible, to satisfy. In looking for a way to provide information for ultimate use in audit guidelines experience in traditional audit disciplines was considered. For example, when a new accounting system is set up with users' needs and requirements in mind, fundamental controls are always included. These safeguards are put in place in order to satisfy control objectives. They are the result of years of experience with like systems and have withstood the tests of time and experience.

The session participants found that the transaction flow as described for accounting systems had the essential characteristics of a general computer application system, encompassing the application program, the external events causing inputs, and the output report or other results. Following this line of reasoning, if one chooses a comprehensive set of control objectives and applies them to an application design at all parts of the system flow, i.e., input, processing and output, a carefully designed set of controls will emerge.

The remainder of Section 5 describes the reasoning leading to the session's approach to identifying and classifying security vulnerabilities and controls for application systems.

5.1 Arthur Andersen and Company (AA&Co.) Control Objectives Approach

A workshop participant presented a specific seasoned approach to accounting systems based on control objectives. In a guide [3] directed towards internal auditors studying and evaluating internal accounting controls, AA&Co. identified an approach using control objectives and applied them in a cycle application and "transaction flow" methodology. Briefly stated, the approach considers that no matter how massive and complex an organization's accounting system may be, it can be divided into a "financial planning and control function" (essentially management) and a limited number of interrelated business "cycles".

5.1.1 Business Cycles as an Auditing Framework.

These business cycles represent the grouping of similar economic events that impact and should be reflected in an entity's financial statement. AA&Co. has defined the following cycles:

- * treasury
- * expenditure (purchasing & payroll)
- * conversion
- * revenue
- * financial reporting.

The use of cycles, in the context of which application programs would be reviewed, provides a meaningful framework for auditing and studying an organization's business and its accounting processes without being overwhelmed by the details of the systems, procedures, techniques, and processing methods. The use of cycles also emphasizes that business activity is a continuous flow over time and that it categorizes the flow of economic events in a logical manner and provides a common basis for discussing what happens in an entity, how to control what happens, and how economic events can impact several segments of the organization at the same time.

5.1.2 Accounting System Control Objectives.

The control objective approach was developed based on the four broadly stated control objectives set forth in accounting literature and in The Foreign Corrupt Practices Act. These objectives cover (a) appropriate authorizations, (b) appropriate accounting classification, (c) substantiation and evaluation and (d) adequate physical safeguards. From these, a set of system control objectives was developed to provide a framework for developing more specific cycle control objectives to be used in the accounting application areas.

The AA&Co. guide enumerates the accounting system control objectives as follows [see [3], pages 46-47]:

- * Appropriate Authorizations. Authorizations should be in accordance with criteria established by the appropriate level of management.

- * Appropriate Accounting Classification. Transactions should be classified (or categorized) in a manner that permits the preparation of financial statements in conformity with generally accepted accounting principles, and management's plan.

- * Substantiation and Evaluation. Report and data base contents should be periodically substantiated and evaluated.

- * Adequate Physical Safeguards. Access to assets should be permitted only in accordance with management's authorization.

- * Recognition of Economic Events. Economic events should be recognized and submitted for acceptance on a timely basis.

- * Acceptance of Transactions. All economic events meeting management's criteria, and only those, should be accurately converted to transactions and accepted for processing on a timely basis.

- * Integrity of Processing. All accepted transactions should be processed accurately, in accordance with management's policies, and on a timely basis.

- * Integrity of Reports. The results of processing should be reported accurately.

- * Integrity of Data Bases. Data base elements should accurately reflect the results of processing.

- * Integrity of Interfaces. Events affecting more than one system should result in transactions that are reflected by each system in the same period.

5.1.3 Cycle Control Objectives.

These system control objectives are used by AA&Co. to develop more specific cycle control objectives for use in reviewing and evaluating internal accounting controls in accounting application areas. This has the distinct advantage of developing consistent and coordinated control objectives for each business cycle application area. The AA&Co. guide lists 117 illustrative cycle control objectives covering typical application cycles in a manufacturing company. Similar cycle control objectives can be readily developed for other industries and organizations. An appendix to the AA&Co. guide lists each of the cycles for a manufacturing company. Within each cycle area, the specific cycle control objectives are identified and for each objective the following discussion is included:

- * Management criteria to be used in conjunction with the objective.
- * Examples of risks if the objective is not achieved.
- * Examples of internal control techniques to achieve the objective.

In the expenditure cycle for payroll, for example, objective 1 is "employees should be hired in accordance with management's criteria." The appendix then lists twelve examples of management criteria, two categories of risks and six types of control techniques. The examples of control techniques include both manual and computer controls.

5.1.4 Advantages of the Control Objectives Approach.

By formalizing control objectives, top management is providing statements of policy which help create a control-oriented environment. These statements are broad, yet sufficiently specific to allow designers and users the ability to set more specific statements or criteria for development of controls. These objectives serve two purposes: (a) They provide clear management guidance for the detailed design of the application systems; and (b) they provide auditors with a set of criteria against which to review specific controls in order to assure that management policy has been carried out.

9-18

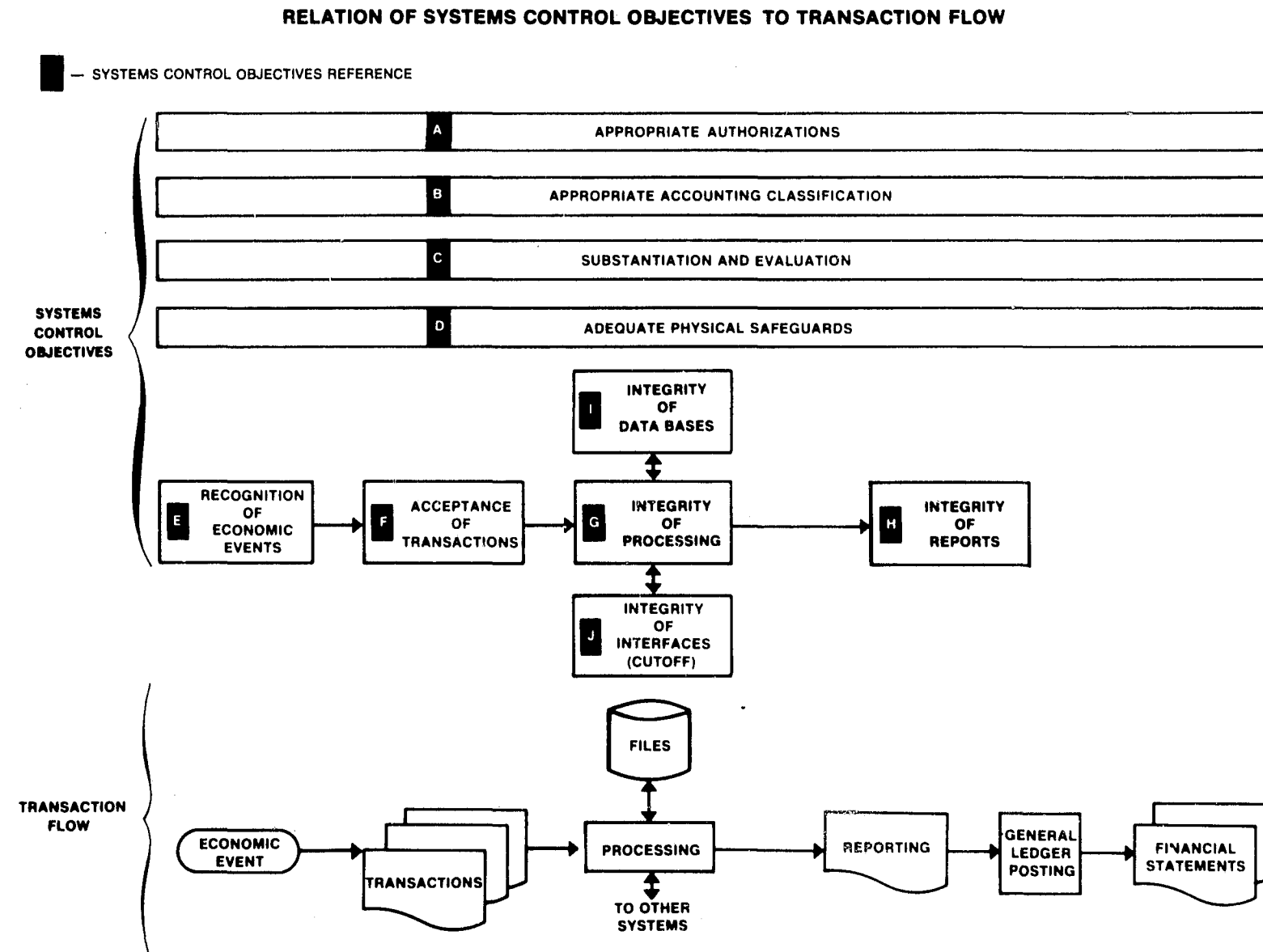


FIGURE 2 RELATION OF SYSTEMS CONTROL OBJECTIVES TO TRANSACTION FLOW (FROM FIG. 5, [3])

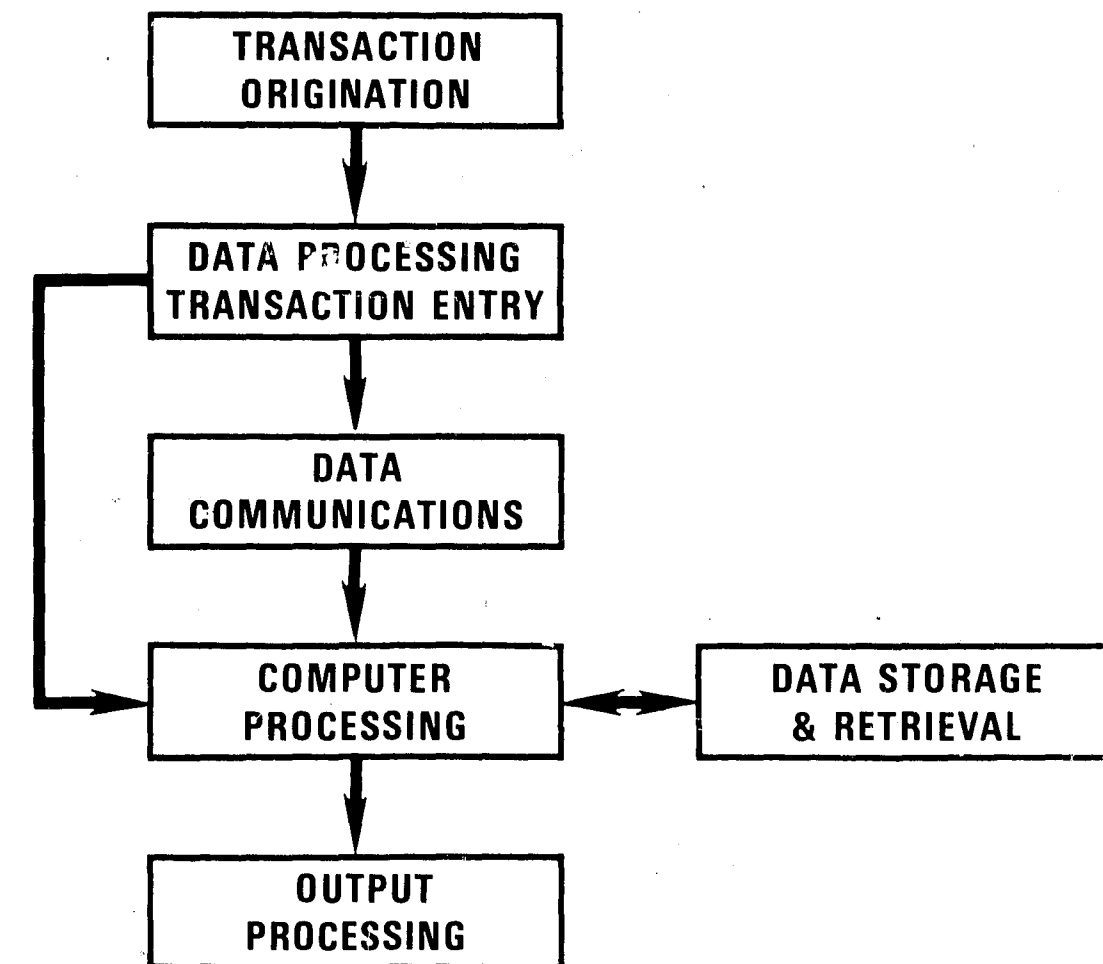
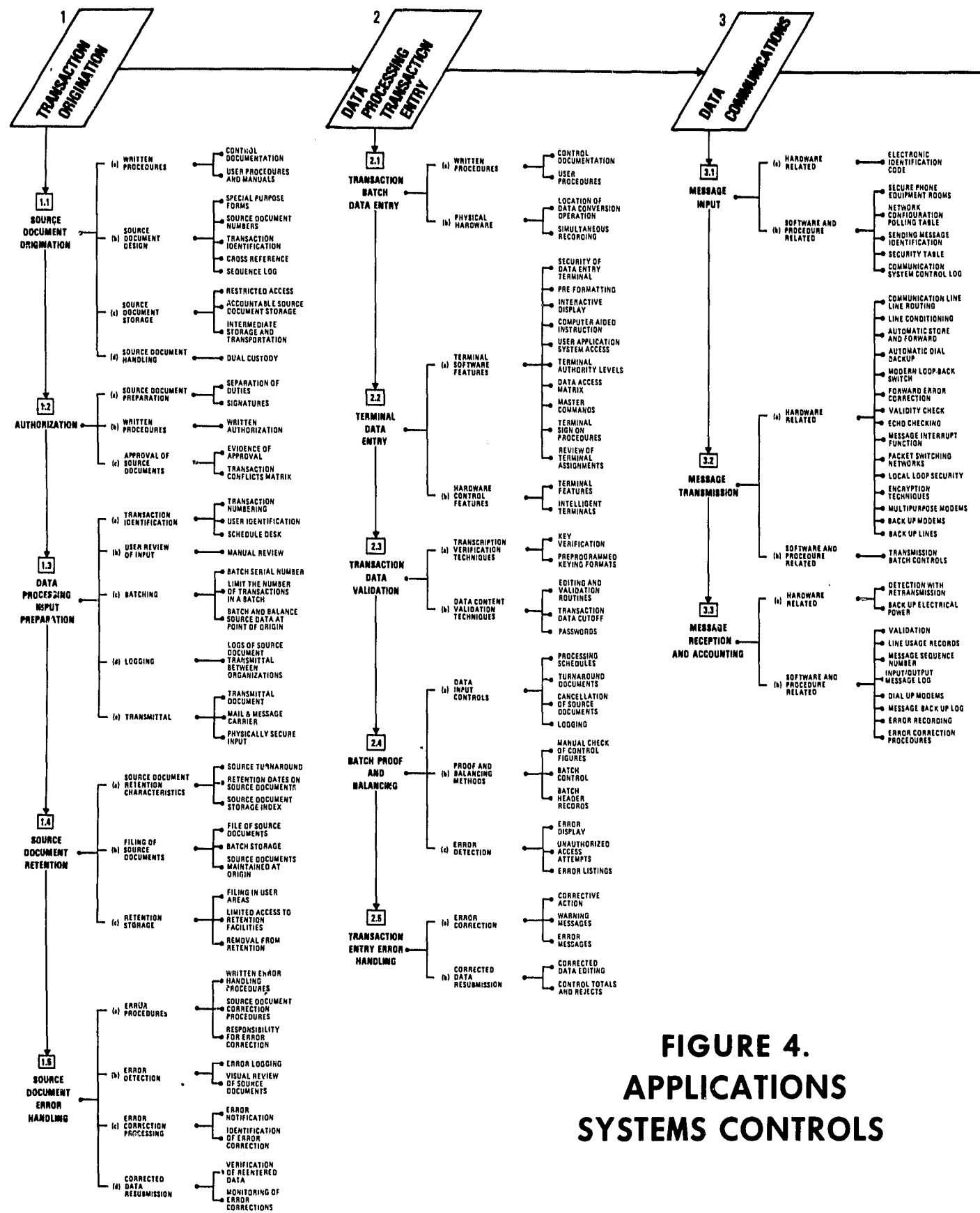
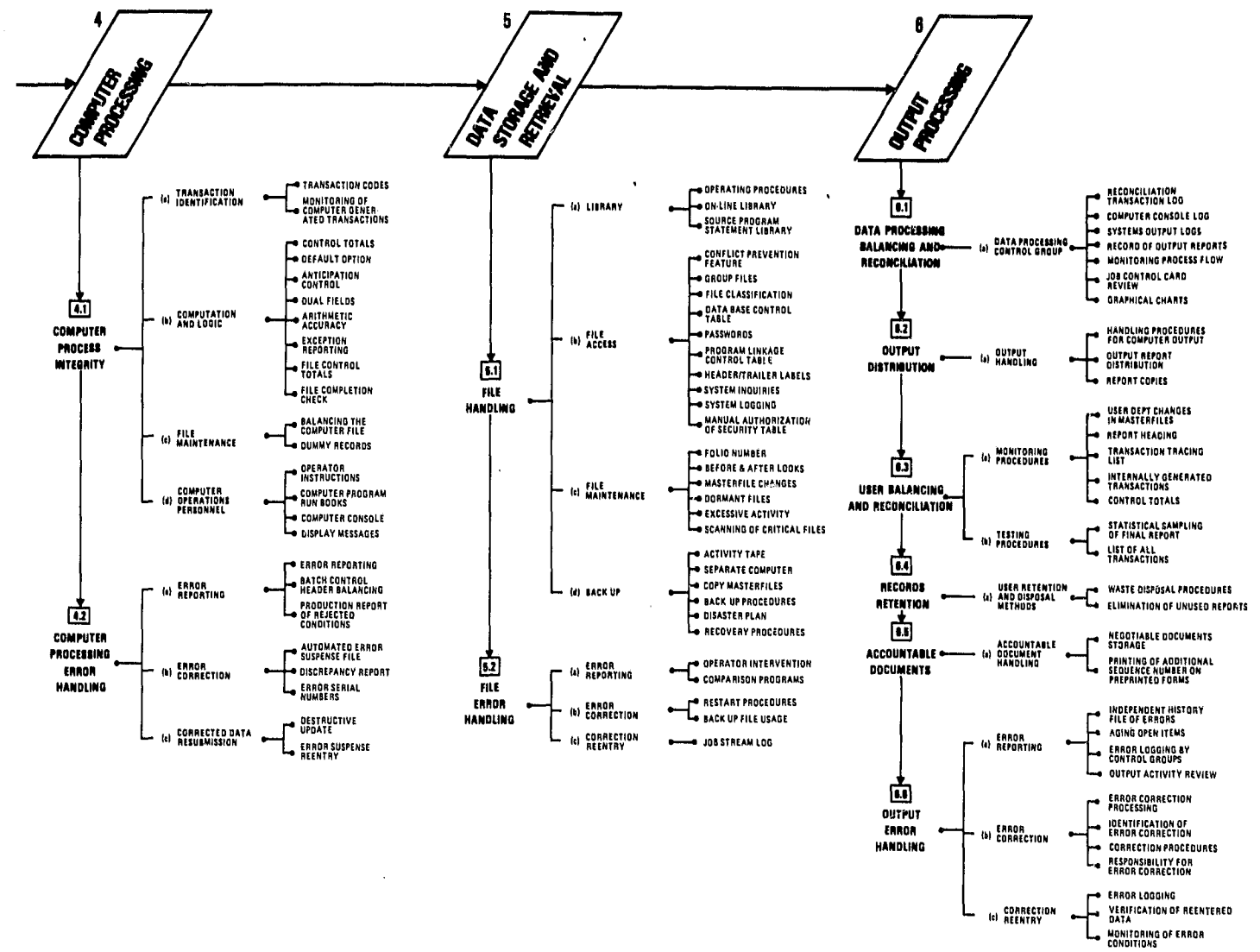


FIGURE 3. SRI TRANSACTION FLOW (FROM FIG. 3-5, [4b])



**FIGURE 4.
APPLICATIONS
SYSTEMS CONTROLS**



SOURCE
PREPARED APRIL 1977 BY THE ADP GROUP FINANCIAL AND
GENERAL MANAGEMENT STUDIES DIVISION, U.S. GENERAL
ACCOUNTING OFFICE FOR ITS AD 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

In addition to providing an approach to controls based on good practices, the breadth of the AA&Co. approach serves to solve the problem of considering the application system in isolation. Adjacent important considerations are well defined.

5.2 Transaction Flow

Figure 2, taken from the Arthur Andersen Guide, illustrates the correlation between the AA objectives and transaction flow. Workshop participants noted the similarity of this representation of transaction flow path to the scheme for classification of application system controls used in the SRI Systems Auditability and Control Study[4].

Figure 3 shows the SRI transaction flow which includes: transaction origination, data processing transaction entry, data communications, computer processing, data storage and retrieval, and output processing. For each flow component SRI further divides the problem into functional (control) areas; control "types"; and finally controls within each control type.

In general, the basic control breakout as defined by SRI is [see [4b], p. 45-46]:

- "* Transaction Origination. Application controls governing the origination, approval, and processing of source documents, the preparation of data processing input transactions, and associated error prevention, detection, and correction procedures.
- * Data Processing Transaction Entry. Application controls governing both remote terminal and batch data entry, data validation, transaction or batch proofing and balancing, error identification and reporting, and error correction and reentry.
- * Data Communications. Controls governing the accuracy and completeness of data communications, including message accountability, data protection, hardware and software, security and privacy, error identification, and reporting.
- * Computer Processing. Application controls governing the accuracy, correctness, and completeness of transaction processing, including transaction validation against masterfiles, error identification and reporting.
- * Data Storage and Retrieval. Application controls to ensure masterfile data accuracy and completeness, correct transaction/masterfile cutoff, data security and privacy, error handling, and backup, recovery, and retention. Note that file integrity controls reflect the growing use of general-purpose file handling and data base software, and an attendant trend to view processing procedures as independent of data files.
- * Output Processing. Application controls governing manual balancing and reconciliation of data processing input and output (both within the data processing input/output control section and at user locations), distribution of data processing output, control over negotiable documents (both within data processing and user areas), and output data retention."

5.3 An Approach Towards Using Control Objectives and Transaction Flow

To provide an overview of all controls contained in the SRI study for each phase of the SRI transaction flow, the U.S. General Accounting Office staff has categorized and named them and composed a master chart. See Figure 4.

This chart provides one of the most comprehensive assemblages of controls available. It can be used by both auditors and application system designers.

The auditor can compare an organization's basic control objectives against components of this chart and controls actually in place. The designer can start with the objectives and review this chart to find appropriate controls which would both satisfy the objectives and be implementable within the developing system. The result will be inclusion of controls addressing the commonly-recognized vulnerabilities. This approach may have less risk and cost less than one in which an attempt is made to identify all vulnerabilities and control them specifically.

5.4 Additional Considerations

Two additional components will help make this strategy most effective. These are: use of auditors in the design stage, and use of an independent review team of experts before implementation.

The auditor has the best opportunity to assure proper development of control systems if involved during the system development phase. If the involvement is limited to reviewing for inclusion of controls as system development progresses and suggesting changes on a timely basis, the auditor will be best able to maintain objectivity and independence. If the involvement is more as a member of the design team, it would be necessary to have another auditor perform the review as a non-team member. Other interrelationships between audit and design of application systems have been analyzed by the session on Audit Considerations in Various System Environments in the first NBS invitational workshop on "Audit and Evaluation of Computer Security." [See [6], PART VI.]

As the methodology described here does not emphasize risk analysis, use of a team of experts to find weak spots, before system implementation, is mandatory. These experts should have experience in penetrating similar systems and could be used to both review code and system specifications as well as perform in a "tiger team" mode. By convening such a group, changes and additions could be made to the system before modifications become too costly to effect, or are applied too late to prevent fraud or misuse.

6. CONCLUSIONS

This paper has attempted to outline a few different strategies for the development of secure application systems. Each of the approaches discussed in this paper has merits but none is complete, i.e., none address all the problems of application system design, development, maintenance, and operation. However, there are recurrent themes which appear in all of them; and the key to these themes is problem simplification.

Three devices which are used to achieve the goal of problem simplification are:

- * Management's early definition of overall system control objectives -- to be used by the designer to build the network of controls; and by the auditor as a benchmark for assessing operational controls.
- * Partitioning of the problem in terms of system life cycle considerations and transaction flow vulnerabilities to be protected against.
- * Employment of schematics such as matrices and flow charts, once partitions have been defined.

A great amount of work is yet to be done and to this end a workshop dedicated solely to analysis of the application system problem may be useful.

In addition, the entire area of data sensitivity has been totally ignored. With Congress and OMB's increasing emphasis on the design and management of secure computer systems this issue can not remain in "haze" indefinitely.

The definitions provided for sensitive application and sensitive data in Section 2.0 are paraphrases of OMB Circular Number A-71 Transmittal Memorandum Number 1 definitions. They are by no means definitive. It has been left for individual organizations to grapple with the problems of just what constitutes an "application requiring a degree of protection because it processes sensitive data" or "data requiring a degree of protection due to the risk and magnitude of loss or harm which could result from inadvertent or deliberate disclosure, alteration, or destruction". Just what "degree" of protection is needed? Should the civil agencies adopt a DoD data classification scheme? These and similar questions require some serious attention.

In conclusion, it is recommended that NBS more fully develop the approaches outlined above and further the dialogue on secure application design by sponsoring additional meetings in this area. It is also recommended that a Federal-wide working group be convened to look into the entire area of defining data sensitivity.

REFERENCES

- [1] FitzGerald, Jerry, "Internal Controls for Computerized Systems," Jerry FitzGerald & Associates, 506 Barkentine Lane, Redwood City, California, 1978.
- [2] "Security for Computer Applications," NBS Draft Guideline, June, 1978.
- [3] "A Guide for Studying and Evaluating Internal Accounting Controls," Arthur Andersen & Co., January 1978.
- [4] "Systems Auditability and Control Study," Stanford Research Institute Report in Three Volumes, Copyright 1977 by The Institute of Internal Auditors, Inc., 249 Maitland Avenue, Altamonte Springs, Florida 32701 U.S.A. Reprinted with permission.
 - a. Ruder, B., Eason, T. S., See, M.E., Russell, S.H., "Audit Practices."
 - b. Russell, S.H., Eason, T.S., FitzGerald, J.M., "Control Practices."
 - c. Sea, M.E., Eason, T.S., "Executive Report."
- [5] "Guideline for Automatic Data Processing Risk Analysis," NBS Federal Information Processing Standards Publication 65, August 1, 1979, available from National Technical Information Service, Springfield, Va. 22161.
- [6] Ruthberg, Z.G., McKenzie, R.G., Editors, "Proceedings of the NBS Invitational Workshop on Audit and Evaluation of Computer Security," NBS Special Publication 500-19, October 1977, Stock No. 003-003-01848-1, for sale by U.S. Government Printing Office, Washington, D.C. 20402.

APPENDIX A

APPLICATION SYSTEM VULNERABILITIES (Published in FIPS PUB 65, pp 22-27)

A number of situations to which applications systems are vulnerable are listed here, grouped according to common system organizational structures. The list is not intended to be all-inclusive but only to suggest the various kinds of vulnerabilities that may exist in each system.

1. **ERRONEOUS OR FALSIFIED DATA INPUT.** Erroneous or falsified input data is the simplest and most common cause of undesirable performance by an applications system. Vulnerabilities occur wherever data is collected, manually processed, or prepared for entry to the computer.

- Unreasonable or inconsistent source data values may not be detected.
- Keying errors during transcription may not be detected.
- Incomplete or poorly formatted data records may be accepted and treated as if they were complete records.
- Records in one format may be interpreted according to a different format.
- An employee may fraudulently add, delete, or modify data (e.g., payment vouchers, claims) to obtain benefits (e.g., checks, negotiable coupons) for himself.
- Lack of document counts and other controls over source data or input transactions may allow some of the data or transactions to be lost without detection--or allow extra records to be added.
- Records about the data-entry personnel (e.g., a record of a personnel action) may be modified during data entry.
- Data which arrives at the last minute (or under some other special or emergency condition) may not be verified prior to processing.
- Records in which errors have been detected may be corrected without verification of the full record.

2. **MISUSE BY AUTHORIZED END USERS.** End users are the people who are served by the ADP system. The system is designed for their use, but they can also misuse it for undesirable purposes. It is often very difficult to determine whether their use of the system is in accordance with the legitimate performance of their job.

- An employee may convert Government information to an unauthorized use; for example, he may sell privileged data about an individual to a prospective employer, credit agency, insurance company, or competitor; or he may use Government statistics for stock market transactions before their public release.
- A user whose job requires access to individual records in a file may manage to compile a complete listing of the file and then make unauthorized use of it (e.g., sell a listing of employees' home addresses as a mailing list).
- Unauthorized altering of information may be accomplished for an unauthorized end user (e.g., altering of personnel records).
- An authorized user may use the system for personal benefit (e.g., theft of services).
- A supervisor may manage to approve and enter a fraudulent transaction.
- A disgruntled or terminated employee may destroy or modify records--possibly in such a way that backup records are also corrupted and useless.
- An authorized user may accept a bribe to modify or obtain information.

3. **UNCONTROLLED SYSTEM ACCESS.** Organizations expose themselves to unnecessary risk if they fail to establish controls over who can enter the ADP area, who can use the ADP

system, and who can access the information contained in the system.

- Data or programs may be stolen from the computer room or other storage areas.
- ADP facilities may be destroyed or damaged by either intruders or employees.
- Individuals may not be adequately identified before they are allowed to enter ADP area.
- Remote terminals may not be adequately protected from use by unauthorized persons.
- An unauthorized user may gain access to the system via a dial-in line and an authorized user's password.
- Passwords may be inadvertently revealed to unauthorized individuals. A user may write his password in some convenient place, or the password may be obtained from card decks, discarded printouts, or by observing the user as he types it.
- A user may leave a logged-in terminal unattended, allowing an unauthorized person to use it.
- A terminated employee may retain access to ADP system because his name and password are not immediately deleted from authorization tables and control lists.
- An unauthorized individual may gain access to the system for his own purposes (e.g., theft of computer services or data or programs, modification of data, alteration of programs, sabotage, denial of services).
- Repeated attempts by the same user or terminal to gain unauthorized access to the system or to a file may go undetected.

4. INEFFECTIVE SECURITY PRACTICES FOR THE APPLICATION. Inadequate manual checks and controls to insure correct processing by the ADP system or negligence by those responsible for carrying out these checks results in many vulnerabilities.

- Poorly defined criteria for authorized access may result in employees not knowing what information they, or others, are permitted to access.
- The person responsible for security may fail to restrict user access to only those processes and data which are needed to accomplish assigned tasks.
- Large funds disbursements, unusual price changes, and unanticipated inventory usage may not be reviewed for correctness.
- Repeated payments to the same party may go unnoticed because there is no review.
- Sensitive data may be carelessly handled by the application staff, by the mail service, or by other personnel within the organization.
- Post-processing reports analyzing system operations may not be reviewed to detect security violations.
- Inadvertent modification or destruction of files may occur when trainees are allowed to work on live data.
- Appropriate action may not be pursued when a security variance is reported to the system security officer or to the perpetrating individual's supervisor; in fact, procedures covering such occurrences may not exist.

5. PROCEDURAL ERRORS WITHIN THE ADP FACILITY. Both errors and intentional acts committed by the ADP operations staff may result in improper operational procedures, lapsed controls, and losses in storage media and output.

Procedures and Controls:

- Files may be destroyed during data base reorganization or during release of disk space.
- Operators may ignore operational procedures; for example, by allowing programmers to operate computer equipment.
- Job control language parameters may be erroneous.

- An installation manager may circumvent operational controls to obtain information.
- Careless or incorrect restarting after shutdown may cause the state of a transaction update to be unknown.
- An operator may enter erroneous information at CPU console (e.g., control switch in wrong position, terminal user allowed full system access, operator cancels wrong job from queue).
- Hardware maintenance may be performed while production data in on-line and the equipment undergoing maintenance is not isolated.
- An operator may perform unauthorized act for personal gain (e.g., make extra copies of competitive bidding reports, print copies of unemployment checks, delete a record from journal file).
- Operations staff may sabotage the computer (e.g., drop pieces of metal into a terminal).
- The wrong version of a program may be executed.
- A program may be executed using wrong data or may be executed twice using the same transactions.
- An operator bypasses required safety controls (e.g., write rings for tape reels).
- Supervision of operations personnel may not be adequate during non-working hour shifts.
- Due to incorrectly learned procedures, an operator may alter or erase the master files.
- A console operator may override a label check without recording the action in the security log.

Storage Media Handling:

- Critical tape files are mounted without being write protected.
- Inadvertently or intentionally mislabeled storage media are erased. In a case where they contain back-up files, the erasure may not be noticed until it is needed.
- Internal labels on storage media may not be checked for correctness.
- Files with missing or mislabeled expiration dates may be erased.
- Incorrect processing of data or erroneous updating of files may occur when card decks have been dropped, partial input decks are used, write rings mistakenly are placed in tapes, paper tape is incorrectly mounted, or wrong tape is mounted.
- Scratch tapes used for jobs processing sensitive data may not be adequately erased after use.
- Temporary files written during a job step for use in subsequent steps are erroneously released or modified through inadequate protection of the files or because of an abnormal termination.
- Storage media containing sensitive information may not get adequate protection because operations staff is not advised of the nature of the information content.
- Tape management procedures may not adequately account for the current status of all tapes.
- Magnetic storage media that have contained very sensitive information may not be degaussed before being released.
- Output may be sent to the wrong individual or terminal.
- Improperly operating output or post-processing units (e.g., bursters, decollators or multipart forms) may result in loss of output.
- Surplus output material (e.g., duplicates of output data, used carbon paper) may not be disposed of properly.
- Tapes and programs that label output for distribution may be erroneous or not protected from tampering.

6. PROGRAM ERRORS. Applications programs should be developed in an environment that requires and supports complete, correct, and consistent program design, good programming practices, adequate testing, review, and documentation, and proper maintenance procedures. Although programs developed in such an environment will still contain undetected errors,

programs not developed in this manner will probably be rife with errors. Additionally, programmers can deliberately modify programs to produce undesirable side-effects or they can misuse the programs they are in charge of.

- Records may be deleted from sensitive files without a guarantee that the deleted records can be reconstructed.
- Programmers may insert special provisions in programs that manipulate data concerning themselves (e.g., payroll programmer may alter his own payroll records).
- Data may not be stored separately from code with the result that program modifications are more difficult and must be made more frequently.
- Program changes may not be tested adequately before being used in a production run.
- Changes to a program may result in new errors because of unanticipated interactions between program modules.
- Program acceptance tests may fail to detect errors that only occur for unusual combinations of input (e.g., a program that is supposed to reject all except a specified range of values actually accepts an additional value).
- Programs, the contents of which should be safeguarded, may not be identified and protected.
- Code, test data with its associated output, and documentation for certified programs may not be filed and retained for reference.
- Documentation for vital programs may not be safeguarded.
- Programmers may fail to keep a change log, to maintain back copies, or to formalize record keeping activities.
- An employee may steal programs he is maintaining and use them for personal gain (e.g., sale to a commercial organization, hold another organization for extortion).
- Poor program design may result in a critical data value being initialized twice. An error may occur when the program is modified to change the data value--but only changes it in one place.
- Production data may be disclosed or destroyed when it is used during testing.
- Errors may result when the programmer misunderstands requests for changes to the program.
- Errors may be introduced by a programmer who makes changes directly to machine code.
- Programs may contain routines not compatible with their intended purpose, which can disable or bypass security protection mechanisms. For example, a programmer who anticipates being fired inserts code into a program which will cause vital system files to be deleted as soon as his name no longer appears in the payroll file.
- Inadequate documentation or labeling may result in wrong version of program being modified.

7. OPERATING SYSTEM FLAWS. Design and implementation errors, system generation and maintenance problems, and deliberate penetrations resulting in modifications to the operating system can produce undesirable effects in the application system. Flaws in the operating system are often difficult to prevent and detect.

- User jobs may be permitted to read or write outside assigned storage area.
- Inconsistencies may be introduced into data because of simultaneous processing of the same file by two jobs.
- An operating system design or implementation error may allow a user to disable audit controls or to access all system information.
- The operating system may not protect a copy of information as thoroughly as it protects the original.
- Unauthorized modification to the operating system may allow a data entry clerk to enter programs and thus subvert the system.
- An operating system crash may expose valuable information such as password lists or authorization tables.

- Maintenance personnel may bypass security controls while performing maintenance work. At such times the system is vulnerable to errors or intentional acts of the maintenance personnel, or anyone else who might also be on the system and discover the opening (e.g., microcoded sections of the operating system may be tampered with or sensitive information from on-line files may be disclosed).
- An operating system may fail to record that multiple copies of output have been made from spooled storage devices.
- An operating system may fail to maintain an unbroken audit trail.
- When restarting after a system crash, the operating system may fail to ascertain that all terminal locations which were previously occupied are still occupied by the same individuals.
- A user is able to get into monitor or supervisory mode.
- The operating system fails to erase all scratch space assigned to a job after the normal or abnormal termination of the job.
- Files are allowed to be read or written without having been opened.

8. COMMUNICATIONS SYSTEM FAILURE. Information being routed from one location to another over communication lines is vulnerable to accidental failures and to intentional interception and modification by unauthorized parties.

Accidental Failures:

- Undetected communications errors may result in incorrect or modified data.
- Information may be accidentally misdirected to the wrong terminal.
- Communication nodes may leave unprotected fragments of messages in memory during unanticipated interruptions in processing.
- Communication protocol may fail to positively identify the transmitter or receiver of a message.

Intentional Acts:

- Communications lines may be monitored by unauthorized individuals.
- Data or programs may be stolen via telephone circuits from a remote job entry terminal.
- Programs in the network switching computers may be modified to compromise security.
- Data may be deliberately changed by individuals tapping the line (requires some sophistication, but is applicable to financial data).
- An unauthorized user may "take over" a computer communication port as an authorized user disconnects from it. Many systems cannot detect the change. This is particularly true in much of the currently available communication equipment and in many communication protocols.
- If encryption is used, keys may be stolen.
- A terminal user may be "spoofed" into providing sensitive data.
- False messages may be inserted into the system.
- True messages may be deleted from the system.
- Messages may be recorded and replayed into the system ("Deposit \$100" messages).

APPENDIX B

TOWARD ESTABLISHING A SYSTEM OF CONTROLS ON SOFTWARE INTEGRITY

Robert P. Abbott

1. INTRODUCTION

The first NBS Invitational Workshop on Audit and Evaluation of Computer Security [1] established the importance of System Life Cycle controls [13,16,18], software development methodologies (such as Structured Programming), and tools [12,13,15,17] in the development and maintenance of software. These controls, tools, and methodologies are important because they potentially increase the reliance an auditor may place upon the applications software. This question of degree of reliance is important for two reasons:

1. The amount of substantive testing [19,20] which an auditor must perform (and hence the cost and duration of the audit) is affected by the degree of reliance which the auditor places upon the application software.
2. Substantive testing is not effective in detecting unauthorized disclosure of information.

The second item above is important and inadequately recognized. Organizations have a vital interest in preventing unauthorized disclosure of information. This interest includes [1,21]:

1. Proprietary and trade secret information concerning the organization's plans, products, and services.
2. Information related to individual privacy (e.g. Privacy Act of 1974).
3. National defense information.
4. Proprietary information which the organization has agreed not to disclose (such as leased commercial software).

It is quite possible to steal information from a computer system and leave no trace whatsoever. Therefore, it is important to have application software upon which the organization can place a high degree of reliance.

In order to place a high degree of reliance upon software, that software must be [1]:

1. Correct
2. Robust
3. Trustworthy

A trustworthy program is one that is well documented, functionally not complex, modular, relatively short in length, integrated into a rigorously structural architecture, and produced as the result of good programming

practices and sensible standards [1].

A program may be correct and robust without being trustworthy. Evidence of a program's trustworthiness includes the presence of good programming practices and sensible programming standards. Various findings from research in Computer Security as well as research in Software Engineering have demonstrated that a lack of good programming practices and sensible programming standards may result in programs which either contain software errors or contain coding structures which are complex in nature. This same research has shown that the existence of errors or complexities are the preconditions which permit the compromise or security violation of application software or system software.

This means that an effective audit of many application systems requires the auditor to perform compliance testing on the software itself to measure the degree to which the organization's programming practices (e.g. structured programming) and coding standards are effective in producing software which has integrity.

2. RESEARCH RESULTS

There are two major areas in which the results of the above mentioned research are of immediate value:

1. Systems for classifying software errors (see paragraph 2.1).
2. A collection of software engineering methodologies which serve to identify, locate and measure adverse coding structures within individual program modules (see paragraph 2.2).

2.1 ERROR CLASSIFICATION SYSTEMS

Three different groups which have had extensive experience in the security evaluation of operating systems have each developed classification systems for programming errors which are found in operating systems [2,3,4,5,6]. Other groups have studied application system errors [15]. Whereas the classification systems differ somewhat, there are many points of similarity. One such point is the belief that the number of error classes is finite, and less than 20 in number.

At least one group [2,3] has successfully transferred and applied the error classification methodology to application systems. This particular error classification scheme identifies 7 error categories:

1. Incomplete validation of parameters
2. Inconsistent validation of parameters
3. Unintended sharing of "sensitive" data
4. Inadequate validation over time
5. Inadequate authorization
6. Violation of limits
7. Exploitable logic error

The first error classification, Incomplete validation of parameters, can, for example, be expanded to provide a set of practical control guidelines:

- 2.1.1. For each module, all incoming parameters must be validated prior to use.

2.1.2. Each module must be checked for:

- i. Presence or absence of parameters
- ii. Data type and format of parameters
- iii. Number and order of parameters
- iv. Value range of parameters
- v. Access rights of calling module(s)

2.2 PROGRAM COMPLEXITY

2.2.1 Software Metrics

Software quality is composed of a number of factors [14]. One factor that relates to integrity is complexity. A number of statistical calculations have been developed which assist in measuring the complexity of software [7,8]. Complexity measurements can point out that the original specification was bad, that the software contains sections of coding which will most likely cause problems during the life cycle of the software, and serve to measure the work product of the programming staff. Measurements of program complexity include, but are not limited to:

- A. Complexity coefficient
- B. Ratio of unique operands to unique operators
- C. Ratio of transfer statements to non-transfer statements

2.2.2 Flow Analysis

Within a program, flow-of-control has to do with the number of GOTOs, IFs, CALLs, and other transfer of control statements or instructions. If, even after structured programming constructs have been applied, the pattern of the flow-of-control is such that it is highly interwoven, then that software is said to be complex [9,10,11,12]. Measurements of flow-of-control complexity include, but are not limited to:

- A. Ratio of backward jumps to total instructions
- B. Number and type of decision instructions
- C. The number of interwoven pathways (i.e. knots).

3. THE ROLE OF THE AUDITOR

It is often the case that the EDP Auditor or EDP Security Auditor does not have a computer background. As such, it will be difficult for this person to personally examine the software or to calculate the statistics. The preceding discussion will insure that the examiner is aware of what the data processing shop should be doing even if the examiner does not know how to do it. As always, the role of the examiner is threefold:

1. To insure that appropriate controls exist.
2. To insure that those controls are in place.
3. To seek evidence that the controls are functioning.

4. REFERENCES

[1] Ruthberg, Z.G., McKenzie, R.G., Editors. "Audit and Evaluation of Computer Security", NBS Special publication 500-19, October 1977.

[2] Abbott, R.P. et al., "Security Analysis and Enhancements of Computer Operating Systems", NBS, NBSIR 76-1041, April 1976.

[3] Konigsford, W.L., "A Taxonomy of Operating-System Security Flaws", Lawrence Livermore Laboratory, UCID-17422, November 1, 1976.

[4] Branstad, D.K., "Privacy and Protection in Operating Systems", Operating Systems Review, ACM SIGOPS, Jan. 1973, pp. 9-17.

[5] Carlstedt, J., et al., "Pattern-Directed Protection Evaluation", USC - Information Sciences Institute, ISI/RR-75-31, June 1975.

[6] Bisbey, R., Hollingworth, D., "Protection Analysis", USC - ISI/SR-78-13, May 1978.

[7] Halstead, M.H., "Elements of Software Science", Elsevier North-Holland, Inc., New York, 1977.

[8] Gilb, T., "Software Metrics", Cambridge, MA, Winthrop, 1977.

[9] Ramamoorthy, C.V. and Ho, S.F., "Testing Large Software with Automated Software Evaluation Systems", in Proc. 1975 Int. Conf. Reliable Software, Los Angeles, Apr 1975, pp 382-394.

[10] Woodward, M.R., et al, "A Measure of Control Flow Complexity in Program Text", Trans. Soft. Eng., Vol. SE-5, No.1, January 1979.

[11] Wetherell, C. and Shannon, A., "Tidy Drawing of Trees", Trans. Soft. Eng., Vol. SE-5, No. 5, September 1979.

[12] Anderson, S.E., and Short, G.E., "A study of Automated Aids for Secure Systems", IBM Data Security and Data Processing, G320-1375, 1974

[13] "Second U.S. Army Software Symposium", Williamsburg, VA, October 1978.

[14] McCall, J.A., et al, "Factors in Software Quality", RADC-TR-77-369, Vol. I (of three), Final Technical Report, November 1977.

[15] Boehm, B.W. et al, "Some Experience with Automated Aids to the Design of Large-Scale Reliable Software", IEEE Tran. On Software Eng., Vol SE-1, No. 1, March 1975.

[16] "Proceedings of the Software Quality and Assurance Workshop", A joint issue by ACM-Performance Evaluation Review (Vol. 7, Nos. 3 and 4) and ACM-Software Engineering Notes, (Vol. 3, No 5), November 1978.

[17] Naughton, J.L., et al, "Structured Programming Series", Vol. XIII, IBM Corporation, Gaithersburg, MD, July 1975.

[18] Glore, J.B., "Software Acquisition Management Guidebook: Life Cycle Events", The Mitre Corporation, Bedford, MA, February 1977.

[19] Mair, W.C., Wood, D.R., Davis, K.W., "Computer Control and Audit", The Institute of Internal Auditors, 1978.

[20] Jenkins, B., and Pinkney, A, "An Audit Approach to Computers", The Institute of Chartered Accountants in England and Wales, 1978.

[21] The EDP Auditor, Vol. 7 No. 2, Summer 1979.

PART X: SESSION 8

Chairperson: Hart Will
University of British Columbia

Participants:

George I. Davida
National Science Foundation

Frank Manola
Computer Corporation of America

Donald Coughlin
Peat Marwick Mitchell & Co.

Frederick Palmer
Palmer Associates

Thomas Fitzgerald
Manufacturers Hanover Trust

David A. Rubin
Peat Marwick Mitchell & Co.

Marvin Schaefer
System Development Corporation



From left to right: Thomas Fitzgerald, Donald Coughlin, Hart Will, Frank Manola, Marvin Schaefer, George I. Davida, (Frederick Palmer absent).

Note: Titles and addresses of attendees can be found in Appendix B.

CONTINUED

2 OF 3

EDITOR'S NOTES

HART J. WILL

Dr. Hart J. Will has been with the Faculty of Commerce and Business Administration at the University of British Columbia since 1969, first as Assistant Professor and currently as Associate Professor of Accounting and Management Information Systems. His research and teaching interests lie in: MIS analysis, design, audit, control and security; data and model base management and administration; audit software in general and ACL (Audit Command Language) in particular. He has worked, consulted, taught and published extensively in Europe and North America.

His activities include: Visiting Research Professor at Gesellschaft fuer Mathematik und Datenverarbeitung (GMD) in Germany, 1974-75; Chairman of U.E.C. International Symposium on Computer Auditing: Legal and Technical Issues, June 18-20, 1975, and Editor of Legal and Technical Issues of Computer Auditing (St. Augustin: GMD and UEC, 1975), the conference proceedings; Associate Editor of INFOR 1975-78; Vice President and Trustee of the EDP Auditors Foundation for Education and Research; Director of Publications and Editor The EDP Auditor since 1978.

His academic degrees are: Diplom-Kaufmann (Free University of Berlin) and Ph.D. (University of Illinois at Urbana-Champaign), and his professional designations are RIA and CDPA.

THE CHARGE TO THE GROUP

Data base management systems can serve as an important element in the implementation of procedures and safeguards for the protection of information. This session was asked to identify the various vulnerabilities of a data base and inherent in the use of the data base management system. The controls that can be employed to counter the identified vulnerabilities were to be addressed. [See PART I, Section 2 for the complete charge given to this group.]

The report that follows is a consensus view of this session.

AUDIT, CONTROL AND SECURITY OF DATA BASE AND DATA BASE MANAGEMENT SYSTEMS

Hart Will, Marvin Schaefer, Frank Manola, Donald Coughlin,
George Davida, Thomas Fitzgerald, Frederick Palmer,

1. INTRODUCTION

The group defined the task environment by means of both an information processing and a security audit framework. These definitions provided the foundation for discussions of several topics in the field of multi-level secure data management. The results were summarized in terms of general control objectives for data base and for data base management systems (DBMS) and in terms of application control objectives. A few recommendations complete this report.

2. DATA BASE ENVIRONMENT

In an attempt to define the data base environment the group identified both an information processing framework and a security audit framework.

2.1 Information Processing Framework

A data base as the depository of symbolic information objects can only be established and used by means of several interfaces. An interface is defined as a channel and a language by means of which two systems can communicate.

Figure 1 depicts this situation and illustrates that access to the data base is via a DBMS which facilitates global (schema) and local (subschema) descriptions of the data base to which users can refer by means of application programs. These are written in any compatible source language and become executable after a translation or interpretation process.

Depending on the users, the language used, and the DBMS available in an organization, one can find numerous forms of the user/language system interfaces and of the language system/data bank interfaces. Likewise, the data base may contain 'live' data for "productive" purposes and 'control' data for "overhead" security, control and audit purposes.

All of these activities are imbedded in an operating system that facilitates the use of a specific machine configuration as illustrated in figure 1. At each of these interfaces, security becomes an issue and we refer to these later as multi-level security considerations.

2.2 Security Audit Framework

To cover current audit considerations in a data base environment, the group considered the security framework illustrated in figure 2. This has been labelled the "component approach" to computer security auditing and is based on the following assumptions.

2.2.1 Management Responsibilities: The establishment and evaluation of the system of controls that relate to computer security is the responsibility of management.

Organizational statements concerning security should be developed by management and published as a written policy. Such policy statements should assign responsibility and

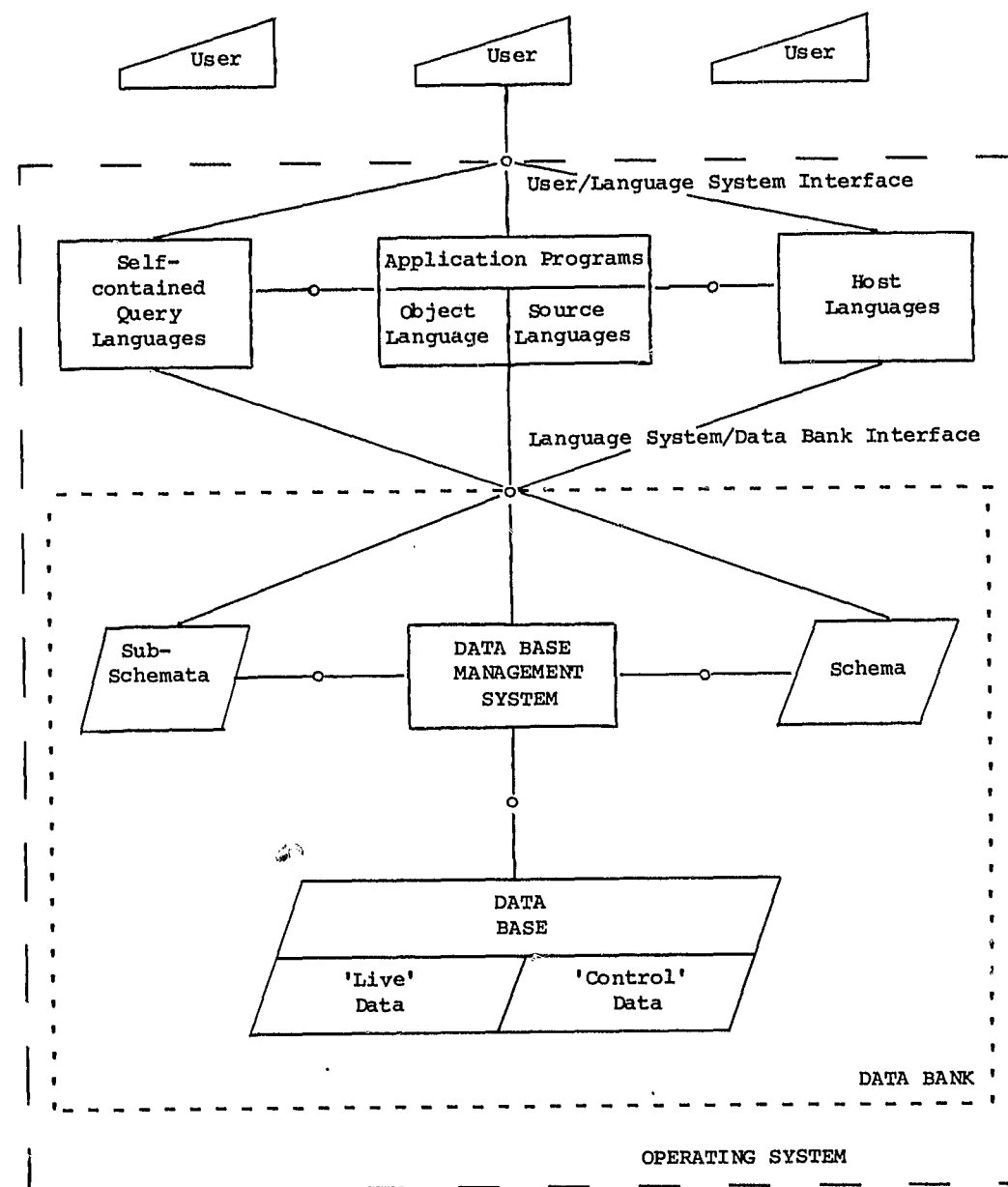


Figure 1: Data Base Environment

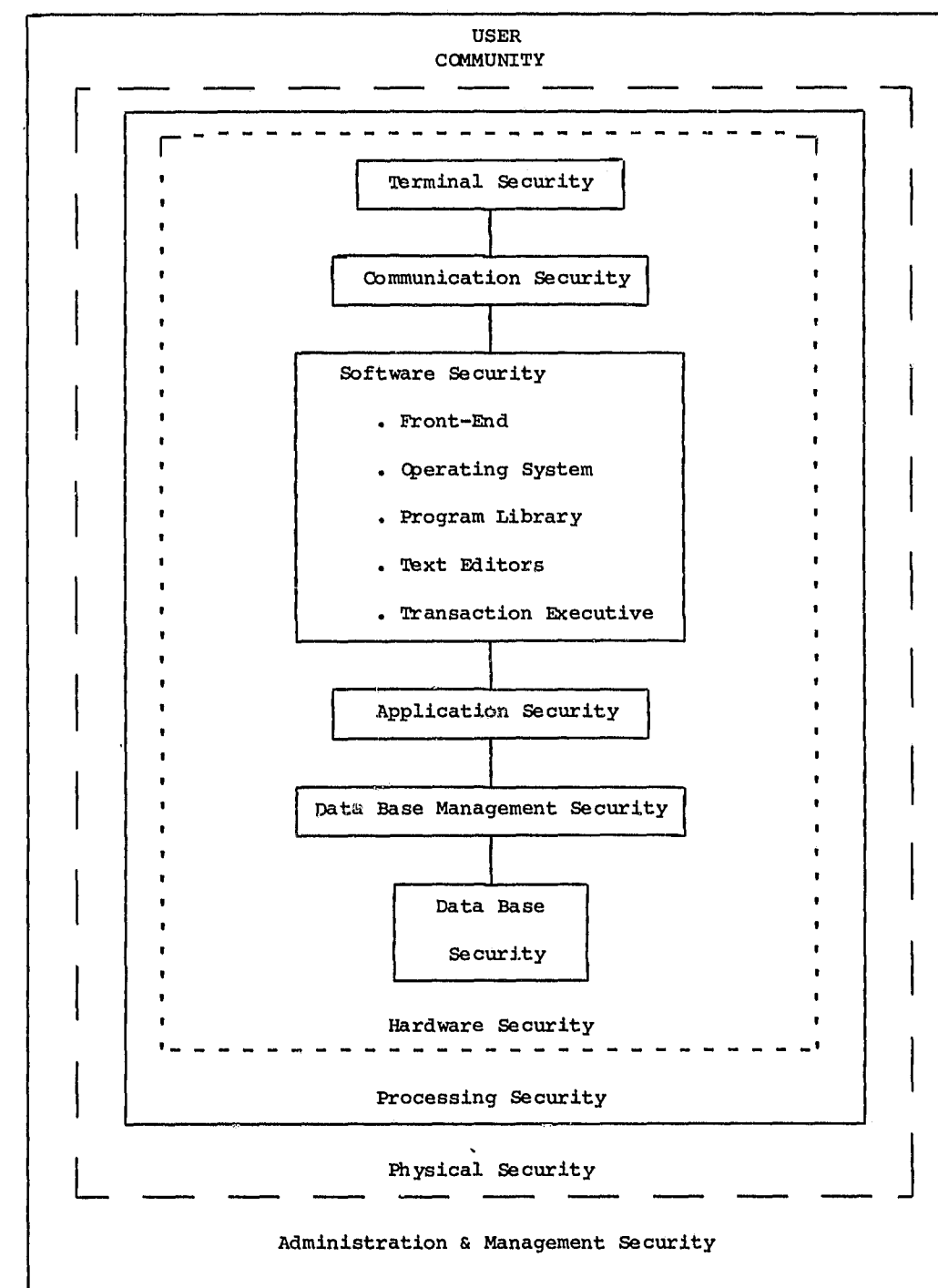


Figure 2: Security Audit Framework

authority for the on-going maintenance of the system of security controls and identify which organizational entities are charged with the test, evaluation, and reporting of control weaknesses to management for corrective action.

The primary audit assumption is that management has instituted the necessary standards, procedures, and operating instructions regarding security within the data processing environment. Should an organization not have instituted such a system of security controls, a management improvement program should be undertaken immediately to implement such a system. While the audit and/or quality assurance function can assist as a participant or reviewer of such a development project, or even execute a "flash review" to determine requirements for the system of security controls, a full scope security audit cannot be undertaken until management has implemented a system of security controls.

The purpose of this assumption is to assist the auditor in reviewing the current level of commitment to security control, to test the controls deemed essential and to formulate recommendations (with accompanying cost/benefit and risk analysis) of the proposed corrective actions being recommended for management action.

2.2.2 Current Technology Constraints: A computer security audit must address the current technology that is being utilized by an organization and the audit must include a comprehensive evaluation of database security when such technology is employed.

A computer security audit must be structured as a total systems evaluation approach and, as such, data base security is but one component to be addressed within the overall scope of the security audit. Data base technology covers a wide spectrum of functions that may overlap other components of the total system to be audited. Therefore it is difficult to formulate a generalized audit work program that includes the appropriate areas of concentration without first understanding the data processing environment that is to be audited.

The component approach to the development of a computer security audit work program is necessary because of the technical knowledge that must be possessed by the audit team in several areas of current technology. An overview of the component approach is shown in figure 2. The areas of control that should be reviewed within a data base management environment and the assessment of the vulnerabilities associated with the lack of such controls must consider the total system evaluation approach since certain controls may be within other components of the system. Where such is the case, appropriate references should be made to the evaluation of other components.

2.2.3 State of the Art Constraints: As will be pointed out in other areas (telecommunications, software control programs, etc.), current data base management systems were not designed to provide the level of security that would protect against a highly knowledgeable technical saboteur.

For example, one may obtain access by gathering relevant documentation, by an intimate knowledge of the specific DBMS implementation and of the system of security controls that is currently operational within an installation. Penetrations are always possible given the current state of the art and the lack of solid commitment from the data processing systems user community to security. Security is not yet a mandatory feature of hardware, firmware, and software offered by the industry's manufacturers and service organizations.

A risk assessment that focuses on the probability of penetration of the existing computer security system by such an expert saboteur should be accomplished. It would determine whether the vulnerability that is to be overcome justifies that cost of the additional levels of control, or whether certain critical processes or data banks should be removed from a shared computer system environment.

Prior to conducting sophisticated penetration studies, the sensitivity of the data maintained by the organization should be defined by a classification structure to clearly delineate what sub-sets of the total organizational automated information require special handling based upon clear and concise management policies regarding the handling, control and dissemination of each class of data.

3. MULTI-LEVEL SECURITY ISSUES

This portion of the report addresses the issues of multi-level secure data base management systems in the context of restricting access to sensitive data on the part of a user population whose members have been accorded varying levels of authorization. These users may have attained their access rights on the basis of a background investigation of the kind normally associated with DoD security clearances, or by reason of the users belonging to an organization to which specific franchise has been accorded as a result of their functional responsibilities.

It is assumed that these users do not all have the same level of clearance. It is further assumed that there will be users with different levels of clearance concurrently using the same computing resource. If all users are of the same level of clearance at the same time, and all have the same level of authorization, the problem is considered to be one of detecting willful malfeasance on the part of an employee (i.e., it is akin to sabotage) and is best addressed through the use of thresholds on the maximum authority accorded a single user (e.g., a bank might require the concurrent action of two authorized users to transfer an amount in excess of \$10,000,000 to another institution), or through the after-the-fact analysis of audit trails in order to detect the commission of a fraudulent deed. These latter issues are considered to be beyond the scope of the present treatment.

We further assume that some of the data on the system is of significant, noninsurable value such that the risk of compromise outweighs the costs associated with the protection measures described below.

3.1 Implementation Issues Relevant to Secure Data Management Systems

At the risk of seeming redundant, we feel compelled to emphasize the importance of correctness to the overall security provided by the data base management system. The correctness issues range from relatively benign computational errors that might be found within the data processing portions of the system, to the more serious errors that could occur in the information storage and retrieval components, or to the potentially catastrophic errors that may be present in the security enforcement features of the system. Any such error can lead to a compromise of security, provided there is a potential exploiter waiting in the wings for his opportunity to strike. This section deals with issues that could lead to an implementation in which there reside exploitable security flaws.

It will be seen that, because of the vast scope of the problem, we do not attempt to present remedies that can be readily applied, but rather indicate reasons for caution and directions for further research.

3.1.1 The Data Base Management System as an Operating System: Modern data base management systems have grown to perform a large number of functions that are traditionally associated with conventional operating systems. They are designed to support a large number of concurrent users, to interface with externally produced application programs that perform assorted data reduction tasks, and to recognize a varied assortment of file storage organizations. Some data base management systems have been designed to interface with existing operating systems, while others have been built to run on bare hardware. In both cases it can be observed that there is present nearly all of the functionality of an operating system: schedulers, I/O managers, authenticators, virtual storage management, user profiles, accounting facilities, etc.

The size of such systems is immense. They are often comprised of millions of assembly language instructions. Their organization is generally optimized to support the functions of rapid retrieval. If the data base management system has also been designed to support frequent updates to existing data bases, there may have also been specialized intertwinings of data which have lead to a number of subsidiary data bases used by the data base management system to support its implementation of each major user data base. Often these internal data bases are constructed at variance with the "fire walls" of least common mechanism and least privilege which are essential to the design of a secure operating system.

Skilled penetrators have been successful in methodically "breaking" numerous existing operating systems because of the presence of security flaws in their implementation. Hence, unless a data base management system is implemented such that such users cannot employ penetration methods against its host operating system, there is always the direct possibility that users will be able to "take over" control of the host machine and thereby obtain access to whatever portions of the databases they desire.

It has been shown that the "safety" question of operating system security is undecidable [18]. This is a rather negative result in that no general techniques can be developed to prove that a system is secure; however, according to the Law of Requisite Variety specific systems may be designed to be secure against known attacks.

Research has been conducted into the design of secure operating systems and secure data management systems. Such research has included designs for kernelized data base management systems [2] or architectures in which the data management system is designed to interface with a secure (kernelized) operating system [11]. In this domain, kernelization, because of its resulting simplicity, appears to be the strategy for conventional computer architectures with the highest potential payoff because of the overwhelming size of the systems under consideration.

There are no secure data management systems in existence today which protect reliably against threats by the skilled penetrator. We anticipate that there will be implementations of data management systems that provide improved security by running under the control of secure operating systems in the next five years (e.g., under KSOS or KVM/370).

3.1.2 Provision for General Programming Capabilities: Increasingly, there is a need for modern data base management systems to interface with a host of special-purpose application programs. These application programs are not part of the existing data base management system. There are generally provisions for the addition of new application programs. In some systems, users have the capability of writing and compiling their own application programs and then using them to process data retrieved by the data base management system.

These application programs present a potential threat to the security of the data bases, since they can be used as a tool for penetration. We therefore recommend that, in the absence of a certifiably secure data base management system, the addition of new application programs be highly restricted, permitting only the addition of application programs that have been audited or otherwise controlled. We further recommend that the use of compilers on the same machine as that on which the data base management system resides be strictly controlled, since they also can lead to system penetration.

3.1.3 System Extensibility: Just as new application programs are a potential weakness in the security of a data management system, so also are possible new extensions to existing data management systems. It is clear that, even as he incorporates extensions to the system, an interloper acting as a systems programmer can produce code in which he has provided himself (or a confederate) with "trapdoors" which will provide him with subsequent access at will to data bases.

Thus, installations in which there is ongoing systems maintenance activity must use diligence in protecting against the introduction of system modifications that will deny them whatever security their system may have originally provided. To some extent protection can be achieved through the vigilant use of auditing techniques on any new code introduced into the system. However, it must be pointed out that, because of the size of the body of code comprising the base system, it is unreasonable to expect that an auditor will be capable of finding any but the most obvious forms of subversion attempts.

3.2 Threats from Within and Threats from Without

Compromises may occur to the security of a data base as the result of actions performed by an individual who has submitted a job to the computer. He may be a user who has obtained unauthorized possession of a password to the system, or he may be an authorized user who is attempting to misuse his authority. He may attack the security safeguards in the system through any of the means common to penetrators. It must be assumed that he has access to system listings since these are so readily available. (If it is a commercially

available system, the penetrator can generally obtain access to system listings by buying them direct from the manufacturer. If not, he may get a copy of the listings by either surveilling the trash cans in which listings are traditionally discarded, or by bribing an employee at the computer center.) He may attack the system by introducing new applications programs, or by exploiting certain errors which will throw the system into an inconsistent state and thus achieve supervisor state.

The installation is also subject to attack by the system maintenance personnel. Some of these attacks may be implemented through innocent modifications to a seemingly unrelated portion of the system which just happen to incorrectly restore state, or which otherwise happen to modify a system authorization data base in such a way that previously existing safeguards have been subverted.

Lastly, there is the case of downright sabotage or collusion. In such cases, trapdoors or Trojan horses are deliberately planted such that a colleague, operating from a terminal, is permitted covert access to privileged data at will.

While malevolent attacks are to be feared, we emphasize that an exploitable, benignly placed flaw, can be most dangerous. Hence, it is again advised that all modifications to the system must be audited prior to their being incorporated into the main computing environment.

3.3 Security and Inference Problems

We do not address the issues of integrity and validity of data in this section of the paper, but wish to concentrate on security as follows. The systems we are concerned with here will prohibit modifications to data by unauthorized users. They are not expected to be sufficiently sophisticated as to prohibit destruction of data base integrity by the so-called authorized user. Hence, there is no "security" issue in permitting an authorized user of a data management system to assign the age "3" to an individual born in 1921, nor in his transferring \$25,000 from an account with a balance of \$12.42. Such controls can be established in various ways by or for authorized users. We identify separate objectives for them below. Security auditors will insist that audit trails be kept on all users and all transactions conducted on the system such that, through the process of applying standard accounting practices, the integrity of the data base can be scrutinized, and as necessary corrected. It is, consequently, essential that the audit and control information be safeguarded by the system such that only authorized users can obtain direct access to it. The level of protection for this data must be at least as high as that for the most sensitive of the protected data bases.

In many database applications protection must be provided both for read as well as write. Writing on the database requires more direct action than reading [4,5]. Reading can often be done indirectly, especially if the database system is used in applications of information retrieval for statistical purposes. In this case techniques have been developed for compromising databases using subject directed queries such as MAX, MIN and AVERAGE [4,5]. For example using MAX or MIN, one can compromise one individual out of n . If n is large, then the bandwidth of compromise may be small. However, in the case of averages, it has been shown that all the individuals involved in the queries can be compromised. Thus the bandwidth of compromise in this case is 100% [5].

Problems of inference do not seem to have a satisfactory solution since this is done using allowable queries and not illegal access! This is an area of intense research [4].

3.4 Audit Trails and User Accountability

Audit trail information is of significant value both to those who need to certify the propriety of an installation's operational procedures, but also to those with the responsibility for identifying the possible commission of a security policy violation.

It is to be noted that an audit trail for a busy installation will eventually comprise a very large data base. In order for auditors to be capable of properly evaluating the contents of this data base, special tools and pattern matching software may be required [24]. The correct functionality of such software is essential and must not be over

looked, since it could be possible for an interloper to rely on flaws in the accounting and audit software as a means of covering up his transgressions. Data gathering mechanisms ought to be implemented as an unmodifiable combination of hardware and firmware, the recording being made into unerasable store. In this way, interlopers will be incapable of destroying evidence of their malfeasance.

Similarly, privacy legislation and evolving national security policy trends are beginning to require that accountability data be maintained for large numbers of transactions for a long period of time. These data bases will be immense and present a significant data management problem in addition to requiring special protection.

3.5 Independent Access for Auditors a Threat

In order that a system provide security, it is necessary that it implement the reference monitor concept. This concept requires that there be security mediation for every access between a subject and an object. The implication of this requirement on conventional computing architectures has been that there be a centralized body of code (often referred to as a "security kernel") which uniquely performs this mediation function.

EDP auditors traditionally require that they be permitted the use of their own auditing tools as one of their checks and balances. They also tend to require that they be permitted to use access paths to data which are independent of the host system that they are auditing. This requirement is levied in order to preclude the possibility that the system misrepresent the actual state of its data bases. The suspicion that the system's access monitor and access routines might be involved in such spoofing provides the justification for their requirement.

However, this requirement introduces a problem, unless based on a common schema processor as suggested by Will [25]. If the auditors are capable of accessing the data by going around the vigilance of the security enforcement mechanism, then there exists a mechanism by which an interloper may obtain access to the data on the system. This could lead to exploitation by members of the user population who are not auditors.

There is certainly no problem in allowing the auditors to have read access to the data bases as they are represented on demountable media (using either a different machine, or using the host machine in stand-alone mode). But the system cannot be set up in such a way as to permit one means of data base access to the general system user, and a completely unmediated form of access to the auditor and still provide any form of certifiable security.

3.6 Possible Data Base Management System Architectures

In this section we will briefly describe possible architectures for secure data management systems. The discussion will be limited to those in which there is:

- i) A secure host operating system;
- ii) A stand-alone data management system with its own security kernel;
- iii) A security mediation function on one computer and a set of stand-alone computers accessible from the first;
- iv) A "secure" subsystem on a computer with a standard operating system; and
- v) A standard data management system with a standard operating system in which the protection is provided through the use of encryption.

These are discussed briefly in the following subsections.

3.6.1 Secure Host Operating System: If one wishes to implement an unprivileged data management system atop an existing secure operating system which supports general usage it is important to note that the implementation cannot provide either more nor less security than that provided by the operating system's security policy. This statement is based on

the fact that the secure operating system will control access to the files containing the classified sensitive data base, and protect it to the level of its classification. If the operating system were to provide means for the user to access the data without making use of the data management system (possible in most operating system architectures), then the user who was not interested in being constrained by added security in the data management system would simply circumvent its controls and go directly after the data by other means. On the other hand, if the data management system tried to downgrade data by extracting data of lower classification from the file of high classification in which the data base was stored, the operating system would interfere with the transfer under the constraints of the "confinement property".

This means that, unless the data management is to contain trusted code (which could operate with the same privilege as the security enforcement mechanism of the operating system), the most one can hope for is a data management system that operates atop a secure file management system. This applies as much to each implementation of a data model (linear lists, hierarchies, networks and relations) as to a "common schema processor" into which all data management functions are funnelled, regardless of the data model preferred by individual users [25].

It has been shown elsewhere [11] that one can implement a multi-level secure relational data management system atop a secure operating system such that the data management system uses no trusted code. However, in so doing one finds that the operating system interferes with the maintenance of data base integrity.

3.6.2 Kernelized Secure Data Management System: To our knowledge, there has been relatively little research performed to date on the design of a kernelized secure data management system [1,2,7]. One data management system was designed to present the relational view of data. It offered the possibility of implementing the more modern features of a data management system, since it controlled all access by users to the data base. To date, no implementation has been funded. Efficiency and integrity considerations were not established.

An ambitious project called DAGS (Datenbank-Grundsystem) is under development at Gesellschaft fuer Mathematik und Datenverarbeitung in Bonn, Germany. DAGS is designed to support various data models and is intended as a secure meta-DBMS [9,19]. Intended as a DBMS machine, DAGS is an interesting project and worth further publicity and discussion.

3.6.3 Back-End Data Management System: In this architecture, there is a small network of computers. The central computer has a secure operating system. From it are connections to a number of dedicated single-level computers, each running its own standard data management system and accessing only its own data bases. The only access to each of these data base management systems is through the computer with the secure operating system.

The main computer serves as a transaction processor. After determining that a user has the right to access one of the data management computers, it processes requests from the user, forwards them to the appropriate "back-end" data management systems, and routes responses to the user's process on the main machine for subsequent processing. While it reduces operational risks and potentially presents multi-level views of data bases, the architecture is vulnerable to a protracted Trojan horse attack.

This architecture has a number of attractive features about it, particularly since none of the data management systems requires any modification. The cost impacts, in addition to the acquisition of additional hardware, are primarily involved with problems associated with maintaining multiple copies of single-level views of data bases that are shared across a set of back-end machines.

3.6.4 Secure Subsystem Approach: The idea behind the secure subsystem approach is to have the host operating system environment so configured that only the data management system may be used. Again, the DBMS may be interpreted as a common schema processor [9,19,25]. This removes alternate access paths to the data base as a threat to security, by limiting the processing environment to the handling of simple user requests. General programming is not permitted. Data base application programs may be permitted, but in such a case they must become part of the data management system, or there must be a transaction

processing executive which handles all requests, forwarding them on to applications programs only when there is security policy justification for doing so.

There are vulnerabilities in such a system. The application programs must be screened carefully prior to incorporating them into the system, lest one contain code designed for penetration. Such a system is also subject to compromise in the event that it contains exploitable errors that could lead to accidental spillage of sensitive data, or to an insecure state that could place an interloper into supervisor state on the computer.

In certain benign environments, the secure subsystem approach is appealing. A good example of what can be done with the secure subsystem approach may be found in Stonebraker's modification to the INGRES System [22]. In this application, the INGRES System maintains a set of user profiles. The profiles contain constraints on the user's authority to access or modify the data base. The constraints are expressed as logical predicates which may be conjoined to each user request, thereby reducing the authority of the user appropriately. Interestingly, since the added conjunctions reduce the scope of the user's view of the data base, they also reduce the amount of searching the data management system must do to respond to the user's request. This is one case wherein security enforcement may have a negative cost associated with it. Similar results have been reported by Fernandes and Grey [10].

3.6.5 Encryption: While access control and inference are discouraging, database encryption is thought (by some) to be a more promising tool in database security [6]. Davida, Wells, and Kam have developed a database encryption system that facilitates access control to the field level. Each field has a read and a different write key. The system has the public key property. Each department, for example, can be given access to the fields that it is supposed to access.

The system facilitates subschema implementation. While the system does not entirely solve the problem of operating system security, it does reduce the amount of code that must be "verified". Finally the system thwarts attacks that depend on pattern matching to determine the presence of certain ciphertext in the database since randomising functions in the system destroy such information.

Encryption provides a means of keeping data bases private from all but those users who have access to the proper keys. However, if the data base is stored in clear form once in the data management system, there is still a possibility of an arbitrary penetrator obtaining access to data while it is being accessed by an authorized user.

Other problems arise with respect to updating the data base (particularly if it is a multi-level data base), or of handling key management issues. (Assignment of keys is similar to assignment of capabilities -- problems of revoking a key once it has been granted to a user, who may have passed it to other users, as well as problems associated with lost keys, have yet to be satisfactorily resolved.

3.7 Data Classification Schemes

In this section we examine several schemes that might be employed for classifying the data in a multi-level data base. In each case, we will comment on any impact such schema may have on implementation mechanisms. Further detail may be found in [11,21].

3.7.1 Global by Data Base: In this scheme, each data base is assigned a classification equal to that of the most sensitive data it contains. This classification scheme can lead to over-classification of most of the data in a data base (we have seen estimates that only three percent of the data in all DoD Top Secret data bases is actually Top Secret). Unless "sanitized" versions of these data bases are also prepared, the costs of providing access to these data bases will include expensive background investigations of all users who will require access to any portion of the data contained therein.

The easiest means of controlling access to such data bases is either through the use of a dedicated machine or through "periods processing" (designation of a period during the day when the machine will only be used for processing of data of some security level; when the machine is initialized as such, only cleared users are permitted to use the machine;

and at the end of the period, the machine is cleared of privileged users and their data, the memory of the machine is "sanitized", a new system is brought up, and service is re-instituted for users of a different degree of clearance). If there is a secure multi-level operating system available, the entire data base may be stored as a file and handled by a data management system operating at the appropriate security level.

3.7.2 Global by Record: In this scheme, each record is classified according to the highest classification level of the data it contains. This results in a true multi-level data base application, and can be implemented using any scheme that will support a multi-level file management system. The data base is partitioned into a collection of mutually disjoint single-level data bases. Each of the single-level data bases is stored as a single-level file (or, if one is using dedicated machines or the back-end architecture, one may store the union of all subdata bases up to and including the classification of the dedicated machine on which it is represented.

The classification of the record is determined at the time of its creation. At the time of its modification, it is possible that its classification may change (e.g., flight plans of aircraft carrying nuclear cargoes may be classified, while all other flight plans are unclassified). The rules for raising classifications of records (i.e., causing them to disappear from the view of users not having sufficient clearance) and lowering classifications of records (causing them to appear into the view of some users) may have complications on account of security policy constraints or because of the possibility of users making inferences as a function of whether they can see a particular record or not.

3.7.3 Global by Field: In a relational data management setting, classification of each field has an appeal. Each collection of fields of the same classification can be stored in a file of that classification along with the key to the record (Note that the classification of the key's fields must not be higher than that of any field in the relation.) It has been shown [2,11] that it is possible to perform all normal data management functions on data bases which are partitioned in this way, provided users are given views consisting of the union of the fields in the relation that are of classification levels lower than or equal to this clearance.

This data base partitioning scheme may be implemented on any computer system with a secure multi-level file management system. It is possible to make such an implementation without encountering the problems of fields appearing and disappearing that were encountered in the previous subsection. (One can create fields with different domains at distinct security levels; e.g., "non-nuclear cargo" at a low level and "nuclear cargo" at a high classification level to take account of the possibilities of "records" having distinct security levels. Lower-level users would never see records with highly sensitive fields in them, while users with high clearances would see the entire virtual relation.)

3.7.4 Privileged Program Controls: A means of permitting users to have limited access to a data base to which they do not have authorization for direct access is to produce a set of privileged (or trusted) sanitization programs which are permitted to access the restricted data base components for the user. These programs might, for example, perform statistical abstracting functions on the data base in those cases where users may see average salaries but not individual salaries, etc.

It is to be observed that certain forms of penetration are possible. Sophisticated users who understand the principles of statistical inference, may be capable of inferring individual salaries, e.g., if permitted to pose a sufficient number of queries over sets of non-void intersection [4,8]. Hence, there may still be a need for a control program that interprets the maximum number of queries a user may pose before being denied further access to the system for some period of time.

3.7.5 Formulary: The formulary is the most general of all forms of data classification scheme. It is a data-dependent (or value-dependent) form of classification. Examples of the granularity to which this form of classification may apply include restrictions such as: "Jones may see the salaries of no more than five individuals who outearn him," "Smith may modify salaries up to \$25,000 by no more than 10%," etc. In the general case there will exist a set of access-control programs associated with the profile of each user.

These programs, or formularies, are invoked for each access the user makes. Some formularies may only make decisions on an access-by-access basis, while others may be used to try to address the aggregation problem (retaining memory over the set of user accesses in order to preclude his gathering sufficient data with which to make unauthorized inferences).

Just as the formulary is the most generalized form of access control, it is also the most difficult to implement. These difficulties arise from the fact that each formulary is a program. Formularies may be written independently of one another yet be mutually dependent or mutually contradictory. Certainly, there is a great dependency on the resolution of program correctness issues before the formulary can be successfully used as a reliable protection mechanism. There are also possibilities that formularies will be susceptible to the same inferential attacks as those in Section 3.7.4 above.

4. CONTROL OBJECTIVES

The group identified a number of control objectives for a data base environment and distinguished general controls as well as application controls. Each of these control objectives is briefly described and it becomes possible to identify the risks incurred or the vulnerabilities evident in case the objective is not met. The group did not have the time to address the risks to any degree of detail and refrains from mentioning illustrative control and audit procedures for the same reason.

4.1 General Control Objectives

The first three objectives were labelled "data base control objectives" and the remaining ten objectives were referred to as "DBMS Control Objectives" during the discussions; however they are presented here as 13 general control objectives. The numbers do not imply a ranking.

4.1.1 Objective #1 - Data Base Access Control: User Access to the data should only be possible through the DBMS and alternative access options - if at all allowed - should be limited to "trusted" software. Sections 3.1, 3.2 and 3.5 provide the rationale for this objective and are therefore not repeated. Section 3.6 offers several architectural solutions to meet this objective.

4.1.2 Objective #2 - Computer Access Control: While a sensitive data base is on-line, it may be necessary to restrict the use of the computer system. Sections 3.1 and 3.2 contain the rationale for this objective and section 3.6 provides a number of approaches to accomplish the objective.

4.1.3 Objective #3 - Software Analysis: Independent software analyses ought to be performed on any software allowed to cohabit with DBMS software. The rationale for this objective is now self-evident.

4.1.4 Objective #4 - Security Profiles: User-specific security profiles ought to be established to identify the access rights to data according to the security policies of the organization. These may exist within the DBMS security module which enforces the security policy of the organization.

Conceptually, a table would exist for each user, defining that user's access rights for any system resource. For example, the table might list:

- (i) programs the user can use,
- (ii) type of transactions the user can enter,
- (iii) data the user can read (e.g., in terms of files, records within files, and/or fields within records),
- (iv) data the user may modify, add, or delete.

Users may also be divided into categories, each category having its own defined

access rights. Such tables may be implemented by having separate tables for each system resource.

4.1.5 Objective #5 - Data Description as Need-to-Know Control: The database ought to be defined/described (DDL) according to the user's information requirements, taking into account principles of need-to-know in the (schema/subschema) definitions. For example, users should be restricted to subschemas which describe only the data they are allowed to access.

Recent database systems incorporate the "user view" or "subschema" concept. This is a user-tailored description of the database that defines the database as the user is to see it. Different users may have different subschemas, depending on the portion of the database the user needs to access. In an environment where the data administrator (or some other central authority) is responsible for defining these subschemas, this can be a powerful security mechanism, since the subschema can be defined so that the user sees only that data that he or she needs to see in terms of the organization's security policy. (Ordinarily, if data is not in a user's subschema, it cannot be accessed by him or her.)

4.1.6 Objective #6 - Data Administration: Data base administration functions should be defined in any organization using a DBMS.

Due to the fact that data are shared by multiple users in a data base environment it is important to establish a data administration function to mediate between the local user interests from a global point of view. The data base administrator establishes many of the controls and is responsible for the security of the data base according to organizational policies. Without this function data chaos may result.

Data are referenced by most users according to a schema or subschema which describes their identification and type. The data administrator is usually responsible for the internal DB structures and guards them against accidental damage by those users who have control over their own subschemas, while protecting the user community at large.

4.1.7 Objective #7 - Control Over Special DBMS Functions: There ought to be control over specialized DBMS functions used by DBA's, security officers and auditors, e.g., a highly-protected tamper-proof access log, special access methods to physical data or use of schema compilers must be secure and auditable.

These functions are very powerful ones because their intended users have special responsibilities not shared by other users within the system. Such functions can be used to change the database definition to add or delete data objects, to add or delete user access rights, or to bypass other security controls to access data on physical devices. As a result, there should be both administrative and computer controls on the use of such functions, as well as logging of use, so as to restrict the application of these functions to the small set of authorized users.

4.1.8 Objective #8 - Control Over Language Use: The use of specific language interfaces (host languages or self-contained query languages) ought to be controlled according to security/privacy considerations.

On a system dedicated to a specific set of applications, only authorized programs should be allowed to run. These programs should have been thoroughly tested, documented and inspected, and should reside on special files, access to which is tightly controlled. Systems which allow simultaneous operating of authorized programs and other programs must contain a mechanism to ensure that no program other than those authorized ones can be used with sensitive data. Tight security is easier to achieve if the authorized programs are separated physically, e.g. with separate operational and development machines.

Systems where all users are restricted in their ability to use the system to predictable functions (e.g. to a fixed set of transactions implemented by tested and inspected programs, or to an inspected high level query language interface) can generally be made more secure than systems in which some users have general programming capabilities. Testing of programs against production databases should be prohibited, and use of maintenance programs and utilities strictly controlled.

4.1.9 Objective #9 - Validity Controls: There ought to be data accuracy assurance of database contents at all times, or a methodology established for discovering anomalies (validity), e.g., database contents vs. schema constraints, database values vs. other database values should be checked.

Many simple validity checks may be made on data to reduce this risk (and often are today by edit programs). Newer database systems allow more central definition of validity constraints in the database definition. Types of constraints include simple value checks (e.g., the sex field must be male or female), relationship checks (e.g., the subordinate to a department record must be an employee), and checks on related values (e.g., for a department and employee to be related, the dept # in the employee must match the dept # in the department).

4.1.10 Objective #10 - Data Sharing Controls: There ought to be protection against anomalies resulting from concurrent usage of the same data, e.g., deadlock, lost updates.

This objective is frequently called integrity control. Two typical problems are well-known. The first is referred to as the "lost update". Suppose two programs both retrieve the same database record, both add something to one of the data items in that record, and then both return a modified record to the database. Unless special provisions have been made for this situation, whichever of the modified records reaches the database first will be over-written by the other one, with the result that the update performed by one of the programs will have been "lost". If this happens, the database is likely to be inaccurate, and may also be inconsistent with defined validity constraints.

The second typical problem has been referred to as that of "inconsistent analyses" [23]. If a program tries to produce an analysis of some part of a database, and if the contents of that part are constantly changing as it works, then in general the consistency and usefulness of the analysis cannot be guaranteed. Suppose, for example, that a program is computing the total balance of a number of bank accounts by reading the relevant records and accumulating a total. Suppose that at the same time a second program transfers an amount from an account the balance of which has been included in the total to one which the first program has not yet read. The result is that the total produced will be meaningless, since it will not represent any state either of the real world or of the database at any definable point in time.

A common solution to these problems is "locking". When the effective operation of a program depends on some portion of the database remaining unchanged, the program requests a "lock" on that portion. Once granted, DBMS guarantees that other concurrent programs may not access that locked portion of the database. Unrestricted locking however, creates the possibility of "deadlock". The simplest example of deadlock occurs when two programs, each having locked a part of the database, are each suspended waiting to extend that part to include part of the database locked by the other. There are well-known means of recovering from deadlock, or restricting locking so that it cannot occur, and if locking is used to control concurrent access, these should be incorporated in the system. Other means of controlling concurrent access have also been proposed for specific environments [DBTG, CCA].

4.1.11 Objective #11 - Consistency Controls: When a database or a view thereof is shared over a set of application programs, the organization's consistency constraints should be uniformly enforced.

Today many validity constraints are enforced by application programs rather than by the DBMS. This can be a problem if the same constraint must be enforced by more than one application, for two reasons:

- i) The several applications may not enforce exactly the same constraint. For example, two applications may both store records with a sex field. One application may check the sex field for male or female values, and not allow blanks, while the other may allow blanks. If the overall installation policy is to allow or not allow personal data without sex information, one of the applications is incorrect.

- ii) The existing applications may correctly apply the constraints, but it may be easy to forget to enforce the constraint in any new applications which may be written in the situation above to store sex data; how can there be assurance that the correct check of the sex field will be included in this new application?

In such an environment, a data dictionary may be useful in making sure that constraints are uniformly enforced. A current trend is toward central enforcement of such constraints by the DBMS, both to avoid the above problems, and because many of such constraints are recognized as being essential parts of the data, independent of the applications which manipulate it. Constraints that continue to be enforced by applications when the DBMS could enforce them should be truly application-dependent ones.

4.1.12 Objective #12 - Recovery Controls: There must be journaling capabilities in the DBMS to support reorganization and recovery functions in accordance with a predefined recovery strategy.

To use a DBMS without adequate recovery capabilities and controls is such a risky affair that a special control objective seems redundant, and yet inadequate support in this aspect has led to extremely time-consuming and costly uses of DBMS. Even if no serious failures occur, usage patterns change and require reorganization.

4.2 Application Control Objectives

To avoid repetitious and redundant enumerations of application controls, this group discussed only two such objectives: application standards and internal audit. Their importance is self-evident and will not be belabored here.

4.2.1 Objective #1 - Application Standards: All applications referring to the database ought to be designed and operated in conformance with organizational policies and standards.

4.2.2 Objective #2 - Internal Audit: Tests for control weaknesses in database applications ought to be performed repeatedly at irregular intervals.

5. RECOMMENDATIONS

Data bases and data base management systems influence the information processing possible in organizations profoundly. The security, control and audit implications can therefore not be separated from the data management support provided by DBMS and operating systems software. As indicated, a number of problems exist and require further study:

5.1 The Role of the National Bureau of Standards (NBS)

NBS should participate in the development and application of criteria for evaluating the "security trustworthiness" of DBMS.

Due to the problems outlined above, this group believes that NBS ought to become actively involved in the establishment of criteria for the design of secure DBMS. The industry is evidently not very interested in a major standardization effort, but that does not mean that NBS could not play the role of a catalyst very successfully. The variety of data base designs and of DBMS is increasing at such a rapid rate that security controls and auditability may be difficult to achieve without a major standardization effort.

As more and more DBMS are being implemented compatibility becomes a problem and a need for standardization is self-evident in order to protect the user community and the legitimate interests of society against the misuse of these systems by making them secure, controlled and auditable.

5.2 Independent Access Paths?

From a security point of view, under current software design technology, there should be no independent access paths to the data provided for special groups such as auditors.

Under secure OS and DBMS architectural designs, auditors should be subject to the same rules and should be restricted to the same access paths to the data as any other user of the data base and the DBMS. This view may be unacceptable in the short run to managements (and to auditors who are paid for their services by companies) due to the required specialization and established "shortcuts", until secure OS and DBMS architectures are designed and implemented.

The consequences of this suggestion could be twofold: A major research effort is to be launched to design secure DBMS architectures before secure OS and DBMS will be demanded and purchased by private industry.

5.3 System Maintenance

There should be further study of the problems of system maintenance in an environment which is to be secure.

Not only the design of a secure DBMS architecture within a secure OS environment is a major area of research, but it is likewise important to address the problems of secure system maintenance. Data base administrators and systems programmers will continuously adapt the data base and its description as well as some of the applications programs to changing conditions and user organizations. Moreover, changes in computer technology will make it essential that whole data bases can be transferred to new machinery without major control, auditability and security problems.

REFERENCES

1. Banerjee, J., Baum, R., and Hsiao, D.K., "Concepts and Capabilities of a Database Computer", ACM Transactions on Database Systems, 3, 4, Dec. 1978.
2. Bonyun, David, Gillian Kirkby and Michael Grohn, "On Specifying the Functional Design for a Protected DMS Tool", ESD-TR-77-140, I.P. Sharp Associates Limited, Ottawa, Canada, March 1977.
3. Chamberlin, D.D., J.N. Gray, and I.L. Traiger, "Views, Authorization and Locking in a Relational Data Base System", Proc. AFIPS National Computer Conference, May 1975, Vol. 44, AFIPS Press, Montvale, N.J., 1975, (also in Database Management Systems, Information Technology Series, Vol. 1, AFIPS Press, Montvale, N.J., 1976).
4. Davida, G., D. Linton, R. Szelag, and D. Wells, "Database Security", IEEE Transactions on Software Engineering, November 1978.
5. Davida, G., J. Kam and D. Wells, "Security and Privacy", Proc. of IEEE COMPSAC, Computer Software and Applications Conference, Chicago, Ill., Nov. 1978.
6. Davida, G., D. Wells and J. Kam, "A Database Encryption System with Subkeys", TR-CS-78, Dept. of EECS, University of Wisconsin, Milwaukee, WI, August 1978, to appear in ACM Transactions on Database Systems.
7. Downs, D., and Popek, G., "A Kernel Design for a Secure Data Base Management System: Proceedings Third International Conference on Very Large Data Bases", October 1977, IEEE Computer Society, Long Beach, CA.
8. Denning, D., "Are Statistical/Databases Secure?", NCC, AFIPS, 1978.
9. Durchholz, R., "Data Models", in H.J. Will (ed.) Legal and Technical Issues of Computer Auditing, Proceedings UEC International Symposium on Computer Auditing: Legal and Technical Issues, June 18-20, 1975, St. Augustin, Germany, (St. Augustin: UEC and GMD, 1975), pp 113-126.
10. Fernandez, E.B., et al, "An Authorization Model for a Shared Data Base", ACM SIGMOD proceedings, May 1975.
11. Hinke, T.H. and Marvin Schaefer, "Secure Data Management System: Final Report", RADC report TR-75-266, System Development Corporation, November 1975.
12. Hoffman, Lance, "The Formulary Model for Access Control and Privacy", Stanford Linear Accelerator Center Report #117, May 1970.
13. Kam, J., J. Ullman, "A Model of Statistical Databases and Their Security", ACM TODS, 1977.
14. Kam, J. and G. Davida, "A Structured Design of Substitution-Permutation Encryption Networks", in Foundations of Secure Computations, Academic Press, 1978.
15. Manola, F.A. and S.H. Wilson, "Data Security Implications of an Extended Subschema Concept", Proc. 2nd USA-Japan Computer Conference, AFIPS Press, Montvale, N.J., 1975, (also in Database Management System, Information Technology Series, Vol. 1, AFIPS Press, Montvale, N.Y., 1976).
16. Martin, J., Security, Accuracy and Privacy in Computer Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973.
17. Merkel, R. and M. Hellman, "Hiding Information in Trapdoor Knapsacks", IEEE Transactions on Information Theory, September 1978.

18. Minsky, N., "Intentional Resolution of Privacy Protection in Database Systems", CACM, Vol. 19 (March 1976), pp. 148-159.
19. Richter, G., "Data Base Management Interfaces", in H.J. Will (ed.) op. cit., pp. 96-112.
20. Rivest, L. Adleman and A. Shamir, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", CACM, Vol. 21 (February 1978), pp. 120-126.
21. Schaefer, Marvin, "On Certain Security Issues Relating to the Management of Data", in The ANSI/SPARC DBMS MODEL, North-Holland Publishing Company, 1977.
22. Stonebraker, Michael and E. Wong., "Access Control in a Relational Data Base Management System by Query Modification", Univ. of Calif. memorandum No. ERL-M438, Berkeley, May 1974.
23. Waghorn, W.J., "The DDL as an Industry Standard?", in Data Base Description, North-Holland Publishing Company, Amsterdam, 1975.
24. Will, H.J. and Group members, "Interactive Audit Tools and Techniques: A Group Consensus Report", in Zella G. Ruthberg and Robert G. McKenzie (eds.), Audit and Evaluation of Computer Security, Washington: NBS, 1977, pp 12.3-12.22.
25. Will, H.J., "Discernible Trends and Overlooked Opportunities in Audit Software", The EDP Auditor, Vol. 6, No. 2, (Winter 1978), pp. 22-45.

APPENDIX A: GLOSSARY

In the course of reading the papers produced by the eight sessions of this workshop, it became increasingly clear that a unified set of definitions of terms, commonly used and not defined in other sources, was needed to clarify the statements being made. Consequently, the Co-Chairpersons of the workshop, with the assistance of other NBS and GAO personnel, drew up a brief list of terms with definitions. This list was circulated by mail among the workshop attendees for comment. The comments received were used to modify the definitions in the original list. The last two terms in this glossary, though defined elsewhere, are included for convenience.

Although the Co-Chairpersons attempted to integrate these comments into the definitions, lack of time prevented any iteration of the comment process. The definitions presented here should therefore be viewed as a step in the right direction but not the consensus view of the workshop. The Editor assumes final responsibility for the content of this glossary.

1. Computer System

A computer system is an interacting or interdependent group of components, consisting of hardware, software, firmware, data, and people functioning as an entity to accomplish a specific set of objectives.

2. Computer Application

A computer application is data (including logically related computer programs) and associated manual activities designed to accomplish specific objectives or functions for the benefit of the computer user.

3. Hazard

A hazard is a chance event of a dangerous nature (natural or man-made) that occurs without design, forethought, or direction, and that can, if it occurs, harm a computer system or facility (e.g., fire, flood, earthquake, accidental unauthorized access to data).

4. Computer Security Flaw

A computer security flaw is an internal defect of a computer system or application, or an unstated capability which deviates from the published specifications of the computer system or application, that can cause unauthorized or inaccurate performance of that system or application.

5. Threat

A threat is a possible event that can, if it occurs, exploit a vulnerability in the security of a computer system or application. Threats include both hazards and the triggering of flaws.

6. Vulnerability

A vulnerability is a design, implementation, or operations flaw that may be exploited by a threat, to cause the computer system or application to operate in a fashion different from its published specifications and to result in destruction or misuse of equipment or data.

7. Susceptibility

Susceptibility is a synonym for vulnerability.

8. Sensitive Application

a) OMB A-71

A sensitive application is a computer application which requires a degree of protection because it processes sensitive data or because of the risk and magnitude of loss or harm that could result from improper operation or deliberate manipulation of the application (e.g., automated decisionmaking systems).

b) A Suggested Refinement

A sensitive application is a computer application which requires a higher degree of protection than that afforded a non-sensitive (or a normal) type computer application because of the risk and magnitude of loss or harm that could result from improper operation or deliberate manipulation of the sensitive application.

9. Sensitive Data

a) OMB A-71

Sensitive data is data which requires a degree of protection due to the risk and magnitude of loss or harm which could result from inadvertent or deliberate disclosure, alteration, or destruction of the data (e.g., personal data, proprietary data).

b) Some Suggested Refinements

In government, data sensitivity to disclosure is solely a function of laws. Data sensitivity to modification is a function of the size of potential benefit to a perpetrator and the size of potential cost to the public.

10. Sensitivity

Sensitivity is the degree of criticality of computer system components to their owners, users, or subjects and is most often established by evaluating the risk and magnitude of loss or harm that could result from improper operation or deliberate manipulation of the component. The components may be hardware, software, firmware, or data.

11. Risk

Risk is the potential loss or damage to an organization, as for example that resulting from the use or misuse of its computer. This may involve unauthorized

disclosure, unauthorized modification, and/or loss of information resources as well as the authorized but incorrect use of a computer. Risk can be measured to some extent by performing a risk analysis.

12. Risk Analysis

Risk analysis is 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.

13. Risk Assessment

Risk assessment is a synonym for risk analysis.

14. Computer Security

a) The current generally accepted version--

Computer security is a state or condition that a computer system possesses. Computer security is never absolute. Rather, each system possesses security at some level. Computer security is provided by internal safeguards (built into the hardware and software) and external safeguards (physical and procedural) against possible threats. The level of computer security is dependent on the degree to which

1) the computer system's components (including hardware, software, firmware, and data) are protected against all significant threats,

2) data maintained on or generated by its data processing systems are accurate and reliable,

3) its data processing systems are operationally reliable and satisfy criteria that assure the accurate and timely performance of the system.

b) A forward looking alternative--

Computer security is a state or condition of security, or resistance to abuse and unauthorized use, that a computer system possesses. Computer security is never absolute. However, each system possesses security to some degree. The degree of computer security is dependent on the degree to which

1) there exists formal policy or security rules for various system components, e.g., human operators, CPU, communications, software, facility.

2) there exists a set of policy enforcement mechanisms that can be trusted to enforce the stated policies and no others.

3) there exists a life cycle system accreditation program which creates and examines technical evidence of the trustworthiness of the enforcement mechanisms to implement the enforcement policies, and reaches a decision on the degree of security to be granted a system based on the risks, threats, and trusted mechanisms.

15. Adequate Computer Security

Adequate computer security is attained when the degree of protection is appropriate to the sensitivity of the data and the cost of recovering from a damaging event, and when all appropriate measures are being used and maintained.

16. Computer Security Audit

A computer security audit is defined as 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.

17. Exposure (dictionary)

Exposure is the condition of being exposed to danger or loss.

18. Firmware (Data Communications Dictionary - Sippl)

Firmware is an extension to a computer's basic command [instruction] repertoire to create microprograms for a user-oriented instruction set. This extension to the basic instruction set is done in read-only memory and not in software. The rom converts the extended instructions to the basic instructions of the computer.

APPENDIX B: WORKSHOP ATTENDEE LIST

This Appendix lists the attendees in alphabetical order, with their titles (where known), affiliations, and addresses. The general format of a listing is as follows with the square brackets indicating the location of the various pieces of information.

Line 1: [Name] [, Workshop Role in addition [Session attended]
to being an attendee]

Line 2: [Job Title and/or Office Title, if known]

Line 3: [Name of Organization]

Line 4,5,... [Address]

Robert P. Abbott (7)
President
EDP Audit Controls
770 Edgewater Drive, Suite 745
Oakland, California 94621

Walter L. Anderson (7)
Associate Director
Financial & General Management
Studies Division
U.S. General Accounting Office
441 G Street, N.W., Room 6011
Washington, D.C. 20548

Robert P. Blanc, Recorder (1)
Chief
Systems and Network
Architecture Division
Institute for Computer Sciences
and Technology
National Bureau of Standards
Washington, D.C. 20234

Sheila Brand, Recorder (7)
Senior Advisor for
Computer Technology
Office of Inspector General
Dept of HEW
Health Care and Systems Review
Rm. 5274 HEW N. Bldg
330 Independence Avenue
Washington, D.C. 20201

Dennis K. Branstad, Recorder (5)
Leader, Computer Systems
Security Group
Operations Engineering Division
Institute for Computer Sciences
and Technology
National Bureau of Standards
Washington, D.C. 20234

Richard Canning, Coordinator
Publisher
Canning Publications Inc.
925 Anza Avenue
Vista, California 92083

LT COL Robert Campbell, Recorder (2)
HQDA (DAMI-AMP)
Room 2E489 Pentagon
Washington, D.C. 20310
Presently
President
Advanced Information Mgmt Inc.
14860 Daytona
Woodbridge, Virginia 22193

P. J. Corum (5)
Mgr, EDP Audit Dept.
Toronto Dominion Bank
Inspection Division
40 University Avenue (S716)
Toronto, Ontario M5W-1P8
Canada
(next page)

Presently
Director Computer Auditing Systems
Pansophic
1651 Old Meadow Rd., Suite 115
McLean, Virginia 22101

Donald T. Coughlin (8)
Principal
Peat Marwick Mitchell & Co.
345 Park Avenue
New York, New York 10022

Howard R. Davia (1)
Director-Office of Audits
[Presently:
Assistant Inspector General
for Audit]
General Services Administration
18th & F Streets, N.W.
Washington, D.C. 20405

George I. Davida (8)
Program Director
Computer Science Section
National Science Foundation
Washington, D.C. 20550
Presently
Univ. of Wisconsin, College of
Engineering & Computer Science
Milwaukee, Wisconsin 53201

Keagle Davis (3)
Partner
Touche Ross & Co.
780 Northstar Center
Minneapolis, Minnesota 55402

Keith O. Dorricott (2)
Partner
DeLoitte, Haskins & Sells
P.O. Box 6, Suite 3630
Royal Bank Plaza, South Tower
Toronto, Ontario M5J 2J1
Canada

Jerry FitzGerald, Chairperson (5)
Principal
Jerry FitzGerald & Associates
506 Barkentine Lane
Redwood City, California 94065

Thomas Fitzgerald (8)
Vice President
Manufacturers Hanover Trust
4 New York Plaza 18/NT
New York, New York 10015

Richard J. Gultinan, Chairperson (3)
Partner
Arthur Andersen & Co.
1345 Avenue of the Americas
New York, New York 10019

David M. Harris (1)
Partner
Lilly & Harris, CPA
1113 Williamson Building
Cleveland, Ohio 44114

Lance J. Hoffman (2)
Associate Professor
George Washington University
Department of Electrical Engineering
and Computer Science
Suite 903
2101 L Street, N.W.
Washington, D.C. 20052

Robert V. Jacobson (4)
President
International Security Tech., Inc.
51 East 42nd Street, Suite 409
New York, New York 10017

Stanley (Stas) Jarocki (2)
ADP Privacy/Security Officer
Bureau of Reclamation
U. S. Dept. of Interior
Engineering & Research Center
P. O. Box 25007
Denver Federal Center
Denver, Colorado 80225

Stephen T. Kent (5)
Research Assistant
MIT Lab for Computer Science
545 Technology Sq., Rm 508
Cambridge, Mass. 02139

Theodore M. P. Lee, Chairperson (6)
Mgr, Systems Security
Sperry UNIVAC [MS 4703]
P.O. Box 43942
Saint Paul, Minnesota 55164

Milton Lieberman (5)
N.E. Sales Manager
Spectron Corp.
49 South Main Street
Spring Valley, New York 10977

Aileen MacGahan (5)
2nd Vice President
Chase Manhattan Bank, N.A.
1 Chase Plaza - 35th Floor
New York, New York 10015
Presently
American Express Company
1 American Express Plaza - 12
New York, N. Y. 10004

Frank Manola (8)
Senior Computer Scientist
Computer Corporation of America
575 Technology Square
Cambridge, Massachusetts 02139

W. Gregory McCormack II (4)
Assistant Auditor-Systems
& President of EDPA
Western Southern Life
400 Broadway
Cincinnati, Ohio 45202
Presently:
Westfield Companies
Internal Audit Department
Westfield Center, Ohio 44251

Robert G. McKenzie, General Co-Chairperson
Audit Manager
Logistics & Communications Div.
U.S. General Accounting Office
441 G Street, N.W., Room 5814
Washington, D.C. 20548
Presently
Director of Audit
Eastern Region
Office of Inspector General
National Aeronautic and
Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

Gerald E. Meyers (3)
Internal Audit Director (34-S)
CNA Insurance
CNA Plaza
Chicago, Illinois 60685
Presently
Executive Vice President
New Faces and Places Inc.
P. O. Box 125
Streamwood, Illinois 60103

Bryan B. Mitchell (1)
Acting Deputy Inspector General
Department of HEW
Room 5274, North Building
330 Independence Avenue, S.W.

Bryan B. Mitchell (1)
Acting Deputy Inspector General
Department of HEW
Room 5274, North Building
330 Independence Avenue, S.W.
Washington, D.C. 20201

William H. Murray, Chairperson (4)
Senior Marketing Support Admin.
Data Security Support Programs
DP Div., IBM Corporation
1133 Westchester Avenue
White Plains, New York 10604
Presently:
Manager, Data Security Program
IBM Corporation
225 John W. Carpenter Freeway East
P. O. Box 2750
Irving, Texas 75062

Peter G. Neumann (6)
Program Manager
SRI International, J1001
333 Ravenswood Avenue
Menlo Park, California 94025

Eric J. Novotny, Recorder (3)
Senior Associate
Computer Resource Controls
17 W. Jefferson Street, Suite 6
Rockville, Md. 20850
Presently
Comsat
950 L'Enfant Plaza, S.W.
Washington, D.C. 20024

Frederick Palmer (8)
Associate, Palmer Associates
170 Park Avenue
Emerson, New Jersey 07630

Kenneth A. Pollock (3)
Assistant Director for ADP
Financial and General Management
Studies Division
U.S. General Accounting Office
441 G Street, N.W., Rm 6011
Washington, D.C. 20548

Darryl V. Poole (3)
Manager, EDP Auditing
American Can Co.
Internal Audit [1C6]
American Lane
Greenwich, Connecticut 06830

Gerald J. Popek (6)
Associate Professor
University of California-LA
Dept. of Computer Sciences
3532 Boelter Hall
Los Angeles, California 90024

James E. Rife, Recorder (6)
Systems Analyst
Logistics and Communications Div.
U.S. General Accounting Office
441 G Street, N.W.
Washington, D.C. 20548

Harry Robinson (2)
Vice President
Metropolitan Life Insurance Co.
One Madison Avenue
New York, New York 10010

Steven J. Ross (3)
Data Security Officer
Irving Trust Company
1 Wall Street
New York, New York 10015

Robert S. Roussey (7)
Partner
Arthur Andersen & Co.
69 W. Washington Street
Chicago, Illinois 60602

David A. Rubin (5)
Mgr of Communications Engrg
Peat Marwick Mitchell & Co.
345 Park Avenue
New York, New York 10022

Zella G. Ruthberg, General Co-Chairperson
Computer Scientist
Security Audit & Evaluation Group
Operations Engineering Division
Institute for Computer Sciences
and Technology
National Bureau of Standards
Washington, D.C. 20234

Frank S. Sato (1)
Deputy Ass't Sec'y of
Defense (Audit) and Director,
Defense Audit Service
Commonwealth Building
Suite 1200
1300 Wilson Blvd.
Arlington, Virginia 22209
Presently:
Inspector General
Department of Transportation
9210 Nassif Building
400 7th Street, S. W.
Washington, D. C. 20590

Donald L. Scantlebury, Keynoter, Chairperson (1)
Director, Financial and General
Management Studies Division
U.S. General Accounting Office
441 G Street, N.W., Room 6001
Washington, D.C. 20548

Marvin Schaefer (8)
Mgr, Trusted Software
Mail Drop 52-09
R&D, System Development Corp.
2500 Colorado Avenue
Santa Monica, California 90406

Gerald E. Short, Chairperson (7)
Project Manager
TRW Defense & Space Group
One Space Park
Bldg R3, Rm 1050
Redondo Beach, California 90278

Joseph A. Sicken (1)
Director of Audit
Department of Commerce
711 14th Street, NW
Washington, D.C. 20230

D. V. Stavola, Recorder (4)
Program Manager
DP Asset Protection
IBM Corporation
1133 Westchester Avenue
White Plains, New York 10604

George W. Steffen (4)
Principal EDP Audit Specialist
Bank Administration Institute
303 South Northwest Highway
P. O. Box 500
Park Ridge, Illinois 60068

Robert Stone (7)
Mgr, EDP Audit
Uniroyal Inc.
Oxford Management and Research
Center
Middlebury, Connecticut 06745

Peter S. Tasker (6)
Group Leader
The MITRE Corporation
P.O. Box 208
Bedford, Massachusetts 01730

M. Zane Thornton, Host
Director (Acting)
[Presently:
Deputy Director]
Institute for Computer Sciences
and Technology
National Bureau of Standards
Washington, D.C. 20234

Stephen T. Walker (6)
Communications, Command, Control
and Intelligence, DoD
Room 3B252, Pentagon
Washington, D.C. 20301

Richard D. Webb, Chairperson (2)
Manager
Peat Marwick Mitchell & Co.
345 Park Avenue
New York, New York 10022

Clark Weissman (6)
Deputy Mgr, R & D Div. &
Chief Technologist
System Development Corp.
2500 Colorado Ave., M1Dp 5209
Santa Monica, Calif. 90406

Hart Will, Chairperson (8)
Associate Professor
Faculty of Commerce
University of British Columbia
2075 Westbrook Place
Vancouver, British Columbia
Canada V6T 1W5

Carl C. Williams (2)
Director-Information Mgmt [185]
American Can Co.
American Lane
Greenwich, Connecticut 06830

Lt.Col. Malcolm L. Worrell (4)
Staff Auditor
Air Force Audit Agency/SWX
Andrews Air Force Base
[Bldg 3802, Stop 22]
Washington, D.C. 20331

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET		1. PUBLICATION OR REPORT NO. SP 500-57	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE COMPUTER SCIENCE & TECHNOLOGY: Audit and Evaluation of Computer Security II: System Vulnerabilities and Controls. (Proceedings of the NBS Invitational Workshop, Miami Beach, FL. Nov. 28-30, 1978.)			5. Publication Date April 1980	
			6. Performing Organization Code	
7. AUTHOR(S) Zella G. Ruthberg, Editor			8. Performing Organ. Report No.	
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, DC 20234			10. Project/Task/Work Unit No.	
			11. Contract/Grant No.	
12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP) same as 9.			13. Type of Report & Period Covered Final	
			14. Sponsoring Agency Code	
15. SUPPLEMENTARY NOTES Library of Congress Catalog Card Number: 80-600034 <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.				
16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.) The National Bureau of Standards, with the support of the U.S. General Accounting Office, sponsored a second invitational workshop on computer security audit, entitled "Audit and Evaluation of Computer Security II: System Vulnerabilities and Controls," in Miami Beach, Florida, on November 28-30, 1978. A cross-section of highly qualified people in the computer science and EDP audit fields was assembled to develop material that would be directly usable for a Federal Information Processing Standard (FIPS) Guideline on the subject. In order to cover the material in a systematic fashion, the workshop was partitioned into three management sessions and five technical sessions. The management sessions addressed Managerial and Organizational Vulnerabilities and Controls at the Staff Level (1 session) and the Line Level (2 sessions). The technical sessions addressed vulnerabilities and controls in the areas of Terminal and Remote Peripherals, Communication Components, Operating Systems, Applications and Non-Integrated Data Files, and Data Base/Data Base Management Systems. These Proceedings are the reports developed by the eight sessions of the workshop.				
17. KEY WORDS (six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons) Applications controls; computer vulnerabilities; data base controls; data base management systems controls; EDP audit; internal audit; operating system controls; system controls; system vulnerabilities; terminal controls.				
18. AVAILABILITY <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input checked="" type="checkbox"/> Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402 <input type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161		19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED	21. NO. OF PRINTED PAGES 210	
		20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	22. Price \$6.00	

USCOMM-DC

ANNOUNCEMENT OF NEW PUBLICATIONS ON
COMPUTER SCIENCE & TECHNOLOGY

Superintendent of Documents,
Government Printing Office,
Washington, D. C. 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)

There's
a new
look
to...

DIMENSIONS

... the monthly magazine of the National Bureau of Standards. Still featured are special articles of general interest on current topics such as consumer product safety and building technology. In addition, new sections are designed to ... PROVIDE SCIENTISTS with illustrated discussions of recent technical developments and work in progress ... INFORM INDUSTRIAL MANAGERS of technology transfer activities in Federal and private labs. ... DESCRIBE TO MANUFACTURERS advances in the field of voluntary and mandatory standards. The new DIMENSIONS/NBS also carries complete listings of upcoming conferences to be held at NBS and reports on all the latest NBS publications, with information on how to order. Finally, each issue carries a page of News Briefs, aimed at keeping scientist and consumer alike up to date on major developments at the Nation's physical sciences and measurement laboratory.

(please detach here)

SUBSCRIPTION ORDER FORM

Enter my Subscription To DIMENSIONS/NBS at \$11.00. Add \$2.75 for foreign mailing. No additional postage is required for mailing within the United States or its possessions. Domestic remittances should be made either by postal money order, express money order, or check. Foreign remittances should be made either by international money order, draft on an American bank, or by UNESCO coupons.

Send Subscription to:

NAME-FIRST, LAST	
COMPANY NAME OR ADDITIONAL ADDRESS LINE	
STREET ADDRESS	
CITY	STATE
ZIP CODE	

PLEASE PRINT

- ☐ Remittance Enclosed
(Make checks payable to Superintendent of Documents)
- ☐ Charge to my Deposit Account No.

MAIL ORDER FORM TO:
Superintendent of Documents
Government Printing Office
Washington, D.C. 20402

NBS TECHNICAL PUBLICATIONS

PERIODICALS

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. As a special service to subscribers each issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year. Annual subscription: domestic \$17; foreign \$21.25. Single copy, \$3 domestic; \$3.75 foreign.

NOTE: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

DIMENSIONS/NBS—This monthly magazine is published to inform scientists, engineers, business and industry leaders, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing. Annual subscription: domestic \$11; foreign \$13.75.

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 principal publication outlet for the foregoing data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements 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 Services, 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 Services, Springfield, VA 22161, in paper copy or microfiche form.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

Cryogenic Data Center Current Awareness Service. A literature survey issued biweekly. Annual subscription: domestic \$25; foreign \$30.

Liquefied Natural Gas. A literature survey issued quarterly. Annual subscription: \$20.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: \$30. Please send subscription orders and remittances for the preceding bibliographic services to the National Bureau of Standards, Cryogenic Data Center (736) Boulder, CO 80303.

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, \$300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215



SPECIAL FOURTH-CLASS RATE
BOOK

END